Account Factory guidance ................................................................. 44
Guidance on Subscribing to SNS Topics ........................................... 44
Guidance for KMS keys ................................................................. 45
Configuration update management .................................................. 46
About Updates .............................................................................. 47
Update Your Landing Zone ............................................................. 48
Manual updates ........................................................................... 48
Resolve drift with Repair and Re-register ........................................ 48
Provision and update accounts using automation ................................ 48
Automate tasks ............................................................................ 50
AWS CloudShell and the AWS CLI ................................................ 51
  Obtaining IAM permissions for AWS CloudShell ......................... 51
  Interacting with AWS Control Tower using AWS CloudShell ....... 52
AWS CloudFormation resources ...................................................... 54
  AWS Control Tower and AWS CloudFormation templates ............ 54
  Learn more about AWS CloudFormation .................................... 54
Customize your landing zone ......................................................... 56
  Customize from the AWS Control Tower console ....................... 56
  Automate customizations outside the AWS Control Tower console 57
Benefits of Customizations for AWS Control Tower (CFCT) ............. 57
Additional CFCT examples ............................................................. 58
Customizations for AWS Control Tower (CFCT) overview .............. 58
  Architecture ............................................................................. 59
  Cost ....................................................................................... 60
Component services ....................................................................... 60
  AWS CodeCommit .................................................................... 60
  AWS CodePipeline ................................................................. 61
  AWS Key Management Service .................................................. 61
  AWS Lambda .......................................................................... 61
  Amazon Simple Notification Service ......................................... 61
  Amazon Simple Storage Service ................................................. 61
  Amazon Simple Queue Service .................................................. 62
  AWS Step Functions .................................................................. 62
  AWS Systems Manager Parameter Store ..................................... 62
Deployment considerations ............................................................. 62
Prepare for deployment .................................................................. 62
To update Customizations for AWS Control Tower ......................... 63
Template and source code ............................................................. 63
  Source code ........................................................................... 63
Deploy CFCT ............................................................................... 64
  Prerequisites .......................................................................... 64
  Deployment steps ..................................................................... 64
  Step 1. Launch the stack ......................................................... 64
  Step 2. Create a custom package .............................................. 66
  Update the stack ...................................................................... 66
  Delete a stack set ..................................................................... 67
Set up Amazon S3 as the configuration source ................................. 68
Operational metrics ...................................................................... 69
CFCT customization guide ............................................................. 69
  Code pipeline overview ........................................................... 69
  Define a custom configuration .................................................. 71
  Root OU ................................................................................. 75
  Nested OU ............................................................................. 76
  Build your own customizations ................................................ 76
  Manifest version upgrades ....................................................... 81
Networking ................................................................................. 84
# AWS Control Tower User Guide

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Factory resources</td>
<td>128</td>
</tr>
<tr>
<td>Account Factory Customization (AFC)</td>
<td>129</td>
</tr>
<tr>
<td>Set up for customization</td>
<td>130</td>
</tr>
<tr>
<td>Create a customized account from a blueprint</td>
<td>134</td>
</tr>
<tr>
<td>Enroll and customize accounts</td>
<td>135</td>
</tr>
<tr>
<td>Add a blueprint to an AWS Control Tower account</td>
<td>135</td>
</tr>
<tr>
<td>Update a blueprint</td>
<td>135</td>
</tr>
<tr>
<td>Remove a blueprint from an account</td>
<td>136</td>
</tr>
<tr>
<td>Partner blueprints</td>
<td>136</td>
</tr>
<tr>
<td>Considerations for Account Factory Customizations (AFC)</td>
<td>136</td>
</tr>
<tr>
<td>In case of a blueprint error</td>
<td>137</td>
</tr>
<tr>
<td>Customizing your policy document for AFC blueprints based on CloudFormation</td>
<td>138</td>
</tr>
<tr>
<td>Additional permissions required for creating a Terraform-based Service Catalog product</td>
<td>139</td>
</tr>
<tr>
<td>AWS Control Tower Account Factory for Terraform (AFT)</td>
<td>139</td>
</tr>
<tr>
<td>Prerequisites</td>
<td>140</td>
</tr>
<tr>
<td>Provision a new account</td>
<td>140</td>
</tr>
<tr>
<td>Multiple account requests</td>
<td>141</td>
</tr>
<tr>
<td>Update an existing account</td>
<td>141</td>
</tr>
<tr>
<td>Deploy AFT</td>
<td>142</td>
</tr>
<tr>
<td>AFT overview</td>
<td>144</td>
</tr>
<tr>
<td>Versions supported</td>
<td>146</td>
</tr>
<tr>
<td>Enable feature options</td>
<td>149</td>
</tr>
<tr>
<td>Resources for AFT</td>
<td>150</td>
</tr>
<tr>
<td>Required roles</td>
<td>153</td>
</tr>
<tr>
<td>Component services</td>
<td>155</td>
</tr>
<tr>
<td>AFT account provisioning pipeline</td>
<td>156</td>
</tr>
<tr>
<td>Account customizations</td>
<td>158</td>
</tr>
<tr>
<td>Alternative VCS</td>
<td>162</td>
</tr>
<tr>
<td>Data protection</td>
<td>164</td>
</tr>
<tr>
<td>Remove an account</td>
<td>164</td>
</tr>
<tr>
<td>Operational metrics</td>
<td>165</td>
</tr>
<tr>
<td>Troubleshooting guide</td>
<td>166</td>
</tr>
<tr>
<td>Drift</td>
<td>169</td>
</tr>
<tr>
<td>Detecting drift</td>
<td>169</td>
</tr>
<tr>
<td>Resolving drift</td>
<td>170</td>
</tr>
<tr>
<td>Considerations about drift and SCP scans</td>
<td>170</td>
</tr>
<tr>
<td>Types of drift to repair right away</td>
<td>171</td>
</tr>
<tr>
<td>Repairable changes to resources</td>
<td>172</td>
</tr>
<tr>
<td>Drift and New Account Provisioning</td>
<td>172</td>
</tr>
<tr>
<td>Types of Governance Drift</td>
<td>172</td>
</tr>
<tr>
<td>Moved Member Account</td>
<td>173</td>
</tr>
<tr>
<td>Removed Member Account</td>
<td>173</td>
</tr>
<tr>
<td>Unplanned Update to Managed SCP</td>
<td>174</td>
</tr>
<tr>
<td>SCP Attached to Managed OU</td>
<td>175</td>
</tr>
<tr>
<td>SCP Detached from Managed OU</td>
<td>175</td>
</tr>
<tr>
<td>SCP Attached to Member Account</td>
<td>176</td>
</tr>
<tr>
<td>Deleted Foundational OU</td>
<td>177</td>
</tr>
<tr>
<td>Security Hub control drift</td>
<td>178</td>
</tr>
<tr>
<td>Trusted access disabled</td>
<td>178</td>
</tr>
<tr>
<td>If you manage resources outside of AWS Control Tower</td>
<td>179</td>
</tr>
<tr>
<td>Referring to resources outside of AWS Control Tower</td>
<td>180</td>
</tr>
<tr>
<td>Externally changing AWS Control Tower resource names</td>
<td>180</td>
</tr>
<tr>
<td>Deleting the Security OU</td>
<td>181</td>
</tr>
<tr>
<td>Removing an account from the Security OU</td>
<td>181</td>
</tr>
<tr>
<td>External changes that are updated automatically</td>
<td>183</td>
</tr>
<tr>
<td>Organizations</td>
<td>184</td>
</tr>
<tr>
<td>Video Walkthrough</td>
<td>184</td>
</tr>
</tbody>
</table>
AWS Control Tower User Guide

CloudWatch .......................................................................................................................... 1378
AWS Config .......................................................................................................................... 1378
AWS Key Management Service ............................................................................................ 1379
AWS Lambda ....................................................................................................................... 1379
AWS Organizations ............................................................................................................ 1379
Considerations ...................................................................................................................... 1379
Amazon S3 ............................................................................................................................. 1380
Security Hub ......................................................................................................................... 1380
Service Catalog ................................................................................................................... 1380
IAM Identity Center ............................................................................................................ 1380

Things to Know About IAM Identity Center Accounts and AWS Control Tower ........ 1382
IAM Identity Center Groups for AWS Control Tower ....................................................... 1382
Amazon SNS ........................................................................................................................ 1384
Step Functions .................................................................................................................... 1385
Security .................................................................................................................................. 1386
Data Protection .................................................................................................................... 1386
Encryption at Rest ................................................................................................................. 1387
Encryption in Transit ............................................................................................................ 1387
Restrict Access to Content ................................................................................................. 1387
Identity and access management ...................................................................................... 1387
Authentication ..................................................................................................................... 1388
Access control .................................................................................................................... 1389
Overview of managing access with IAM ........................................................................... 1389
Prevent confused deputy attacks ....................................................................................... 1396
Using identity-based policies (IAM policies) ....................................................................... 1397
Compliance Validation ........................................................................................................ 1407
Resilience ............................................................................................................................. 1407
Infrastructure Security ........................................................................................................ 1407
Logging and monitoring ...................................................................................................... 1409
Monitoring ........................................................................................................................... 1410
Logging AWS Control Tower Actions with AWS CloudTrail ........................................ 1410
AWS Control Tower Information in CloudTrail ............................................................... 1411
Example: AWS Control Tower Log File Entries ............................................................... 1412
Monitoring resource changes with AWS Config ............................................................ 1413
Managing AWS Config costs in AWS Control Tower ...................................................... 1414
View the AWS Config recorder data on enrolled accounts ............................................ 1414
Troubleshooting AWS Config in AWS Control Tower .................................................... 1415
Lifecycle Events .................................................................................................................. 1416
CreateManagedAccount ..................................................................................................... 1418
UpdateManagedAccount ..................................................................................................... 1419
EnableGuardrail .................................................................................................................. 1419
DisableGuardrail ............................................................................................................... 1420
SetupLandingZone ................................................................................................................ 1421
UpdateLandingZone ............................................................................................................ 1422
RegisterOrganizationalUnit ............................................................................................... 1424
DeregisterOrganizationalUnit ............................................................................................ 1424
PrecheckOrganizationalUnit .............................................................................................. 1425
User notifications ................................................................................................................. 1427
Walkthroughs ...................................................................................................................... 1429
Walkthrough: Move from ALZ to AWS Control Tower ................................................... 1429
Walkthrough: Automate Account Provisioning in AWS Control Tower by Service Catalog APIs ............................................................................................................................... 1429
Sample provisioning input for Service Catalog API ......................................................... 1431
Video Walkthrough ............................................................................................................ 1432
Walkthrough: Configure AWS Control Tower Without a VPC ........................................ 1432
Delete the AWS Control Tower VPC ............................................................................... 1432
Create an Account in AWS Control Tower Without a VPC ............................................ 1433
Walkthrough: Set Up Security Groups in AWS Control Tower With AWS Firewall Manager ........................................... 1434
Set Up Security Groups With AWS Firewall Manager ........................................................................................................ 1434
Walkthrough: Decommission an AWS Control Tower Landing Zone .................................................................................. 1434
  Overview of the decommissioning process .............................................. 1435
  Resources not removed during decommissioning .................................. 1435
  How to decommission a landing zone .................................................. 1442
  Setup after decommissioning a landing zone ........................................ 1443
Troubleshooting ..................................................................................... 1445
  Landing Zone Launch Failed .................................................................. 1445
  Landing zone not up to date error ....................................................... 1445
  New Account Provisioning Failed ....................................................... 1446
  Failed to Enroll an Existing Account .................................................. 1446
  Unable to Update an Account Factory Account .................................. 1447
  Unable to Update Landing Zone ........................................................ 1448
  Failure Error that Mentions AWS Config ............................................ 1448
  No Launch Paths Found Error ............................................................ 1449
  Received an Insufficient Permissions Error ......................................... 1450
  Detective controls are not taking effect on accounts ....................... 1450
  Rate exceeded error returned by the AWS Organizations API ............. 1450
  Failure to move an Account Factory account directly from one AWS Control Tower landing zone to another AWS Control Tower landing zone ......................................................... 1451
  AWS Support ..................................................................................... 1452
Related information ............................................................................. 1453
  Tutorials and labs .............................................................................. 1453
  Networking ....................................................................................... 84
  Security, identity, and logging ........................................................... 1453
  Deploying resources and managing workloads .................................. 1454
  Working with existing organizations and accounts ............................. 1454
  Automation and integration .............................................................. 1454
  Migrating workloads ......................................................................... 1455
  Related AWS services ....................................................................... 1455
  AWS Marketplace solutions .............................................................. 1455
Release notes ....................................................................................... 1456
  January 2023 - Present ...................................................................... 1456
    AWS Control Tower available in Asia Pacific (Melbourne) Region .... 1457
    Transition to new AWS Service Catalog External product type ........ 1457
    New control API available ............................................................ 1457
    AWS Control Tower adds additional controls ................................. 1458
    New drift type reported: trusted access disabled ............................ 1459
    Four additional AWS Regions ........................................................ 1460
    AWS Control Tower available in Tel Aviv Region .............................. 1460
    AWS Control Tower launches 28 new proactive controls ................. 1460
    AWS Control Tower deprecates two controls ................................. 1462
    AWS Control Tower landing zone version 3.2 ................................. 1462
    AWS Control Tower handles accounts based on ID .......................... 1463
    Additional Security Hub detective controls available in the AWS Control Tower controls library ................................. 1463
    AWS Control Tower publishes control metadata tables .................. 1464
    Terraform support for Account Factory Customization .................... 1464
    AWS IAM Identity Center self-management available for landing zone .......................................................................... 1465
    AWS Control Tower addresses mixed governance for OUs ............ 1465
    Additional proactive controls available ........................................... 1465
    Updated Amazon EC2 proactive controls ....................................... 1467
    Seven additional AWS Regions available ...................................... 1467
    Account Factory for Terraform (AFT) account customization request tracing ............................................................... 1468
    AWS Control Tower landing zone version 3.1 ................................. 1468
    Proactive controls generally available .......................................... 1469
January - December 2022 ................................................................. 1469
Concurrent account operations .......................................................... 1470
Account Factory Customization (AFC) .................................................. 1470
Comprehensive controls assist in AWS resource provisioning and management ................................................................. 1471
Compliance status viewable for all AWS Config rules .............................. 1471
API for controls and a new AWS CloudFormation resource ....................... 1471
CfCT supports stack set deletion .......................................................... 1472
Customized log retention .................................................................... 1472
Role drift repair available .................................................................... 1472
AWS Control Tower landing zone version 3.0 ........................................ 1473
The Organization page combines views of OUs and accounts .................... 1475
Easier enroll and update for individual member accounts ......................... 1475
AFT supports automated customization for shared AWS Control Tower accounts ................................................................. 1476
Concurrent operations for all optional controls ...................................... 1476
Existing security and logging accounts .................................................. 1477
AWS Control Tower landing zone version 2.9 ........................................ 1477
AWS Control Tower landing zone version 2.8 ........................................ 1477
January - December 2021 ................................................................. 1478
Region deny capabilities .................................................................... 1479
Data residency features ...................................................................... 1479
AWS Control Tower introduces Terraform account provisioning and customization ................................................................. 1479
New lifecycle event available ............................................................. 1480
AWS Control Tower enables nested OUs .............................................. 1480
Detective control concurrency ............................................................ 1480
Two new Regions available .................................................................. 1481
Region deselection ............................................................................. 1481
AWS Control Tower works with AWS Key Management Systems .............. 1482
Controls renamed, functionality unchanged .......................................... 1482
AWS Control Tower scans SCPs daily to check for drift ......................... 1482
Customized names for OUs and accounts .............................................. 1483
AWS Control Tower landing zone version 2.7 ........................................ 1483
Three new AWS Regions available .................................................... 1484
Govern selected Regions only ............................................................ 1484
AWS Control Tower now extends governance to existing OUs in your AWS organizations ................................................................. 1485
AWS Control Tower provides bulk account updates ............................... 1485
January - December 2020 ................................................................. 1485
AWS Control Tower console now links to external AWS Config rules .......... 1486
AWS Control Tower now available in additional Regions ......................... 1486
Guardrail update .............................................................................. 1486
AWS Control Tower console shows more detail about OUs and accounts ........................................................................... 1487
Use AWS Control Tower to set up new multi-account AWS environments in AWS Organizations ................................................................. 1487
Customizations for AWS Control Tower solution ................................... 1488
General availability of AWS Control Tower version 2.3 ............................ 1488
Single-step account provisioning in AWS Control Tower ......................... 1488
AWS Control Tower decommissioning tool .......................................... 1489
AWS Control Tower lifecycle event notifications .................................... 1489
January - December 2019 ................................................................. 1489
General availability of AWS Control Tower version 2.2 ............................ 1490
New elective controls in AWS Control Tower ....................................... 1490
New detective controls in AWS Control Tower ..................................... 1490
AWS Control Tower accepts email addresses for shared accounts with different domains than the management account ................................................................. 1491
General availability of AWS Control Tower version 2.1 ............................ 1491
Document history .............................................................................. 1492
Control metadata tables ...................................................................... 1501
AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED .................................. 1501
AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED ........................................ 1504
AWS Control Tower User Guide
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.NEPTUNE.PR.3</td>
<td>1950</td>
</tr>
<tr>
<td>CT.NEPTUNE.PR.4</td>
<td>1953</td>
</tr>
<tr>
<td>CT.NEPTUNE.PR.5</td>
<td>1955</td>
</tr>
<tr>
<td>CT.NETWORK-FIREWALL.PR.1</td>
<td>1958</td>
</tr>
<tr>
<td>CT.NETWORK-FIREWALL.PR.2</td>
<td>1960</td>
</tr>
<tr>
<td>CT.NETWORK-FIREWALL.PR.3</td>
<td>1963</td>
</tr>
<tr>
<td>CT.NETWORK-FIREWALL.PR.4</td>
<td>1965</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.1</td>
<td>1968</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.10</td>
<td>1971</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.11</td>
<td>1974</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.12</td>
<td>1976</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.13</td>
<td>1979</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.14</td>
<td>1981</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.15</td>
<td>1984</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.16</td>
<td>1986</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.2</td>
<td>1989</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.3</td>
<td>1992</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.4</td>
<td>1994</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.5</td>
<td>1997</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.6</td>
<td>1999</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.7</td>
<td>2002</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.8</td>
<td>2004</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.9</td>
<td>2007</td>
</tr>
<tr>
<td>CT.RDS.PR.1</td>
<td>2009</td>
</tr>
<tr>
<td>CT.RDS.PR.10</td>
<td>2012</td>
</tr>
<tr>
<td>CT.RDS.PR.11</td>
<td>2015</td>
</tr>
<tr>
<td>CT.RDS.PR.12</td>
<td>2018</td>
</tr>
<tr>
<td>CT.RDS.PR.13</td>
<td>2020</td>
</tr>
<tr>
<td>CT.RDS.PR.14</td>
<td>2023</td>
</tr>
<tr>
<td>CT.RDS.PR.15</td>
<td>2025</td>
</tr>
<tr>
<td>CT.RDS.PR.16</td>
<td>2028</td>
</tr>
<tr>
<td>CT.RDS.PR.17</td>
<td>2030</td>
</tr>
<tr>
<td>CT.RDS.PR.18</td>
<td>2033</td>
</tr>
<tr>
<td>CT.RDS.PR.19</td>
<td>2035</td>
</tr>
<tr>
<td>CT.RDS.PR.2</td>
<td>2038</td>
</tr>
<tr>
<td>CT.RDS.PR.20</td>
<td>2041</td>
</tr>
<tr>
<td>CT.RDS.PR.21</td>
<td>2043</td>
</tr>
<tr>
<td>CT.RDS.PR.22</td>
<td>2046</td>
</tr>
<tr>
<td>CT.RDS.PR.23</td>
<td>2048</td>
</tr>
<tr>
<td>CT.RDS.PR.24</td>
<td>2051</td>
</tr>
<tr>
<td>CT.RDS.PR.25</td>
<td>2053</td>
</tr>
<tr>
<td>CT.RDS.PR.3</td>
<td>2056</td>
</tr>
<tr>
<td>CT.RDS.PR.4</td>
<td>2058</td>
</tr>
<tr>
<td>CT.RDS.PR.5</td>
<td>2061</td>
</tr>
<tr>
<td>CT.RDS.PR.6</td>
<td>2064</td>
</tr>
<tr>
<td>CT.RDS.PR.7</td>
<td>2066</td>
</tr>
<tr>
<td>CT.RDS.PR.8</td>
<td>2069</td>
</tr>
<tr>
<td>CT.RDS.PR.9</td>
<td>2071</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.1</td>
<td>2074</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.2</td>
<td>2076</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.3</td>
<td>2079</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.4</td>
<td>2081</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.5</td>
<td>2084</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.6</td>
<td>2086</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.7</td>
<td>2089</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.8</td>
<td>2092</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.9</td>
<td>2094</td>
</tr>
</tbody>
</table>
AWS Control Tower User Guide

CT.S3.PR.1 ................................................................. 2097
CT.S3.PR.10 .............................................................. 2099
CT.S3.PR.11 ............................................................... 2102
CT.S3.PR.2 ................................................................. 2104
CT.S3.PR.3 ................................................................. 2107
CT.S3.PR.4 ................................................................. 2109
CT.S3.PR.5 ................................................................. 2112
CT.S3.PR.6 ................................................................. 2114
CT.S3.PR.7 ................................................................. 2117
CT.S3.PR.8 ................................................................. 2119
CT.S3.PR.9 ................................................................. 2122
CT.SAGEMAKER.PR.1 ................................................. 2124
CT.SAGEMAKER.PR.2 ................................................. 2127
CT.SAGEMAKER.PR.3 ................................................. 2129
CT.SQS.PR.1 ............................................................. 2152
CT.SQS.PR.2 ............................................................. 2154
CT.STEPFUNCTIONS.PR.1 .......................................... 2137
CT.STEPFUNCTIONS.PR.2 .......................................... 2139
CT.WAF-REGIONAL.PR.1 ........................................... 2142
CT.WAF-REGIONAL.PR.2 ........................................... 2144
CT.WAF.PR.1 ............................................................ 2147
CT.WAF.PR.2 ............................................................ 2150
CT.WAFV2.PR.1 ......................................................... 2152
CT.WAFV2.PR.2 ......................................................... 2155
SH.ACM.1 ............................................................... 2157
SH.APIGateway.1 ..................................................... 2159
SH.APIGateway.2 ..................................................... 2160
SH.APIGateway.3 ..................................................... 2162
SH.APIGateway.4 ..................................................... 2163
SH.APIGateway.5 ..................................................... 2165
SH.APIGateway.8 ..................................................... 2166
SH.APIGateway.9 ..................................................... 2168
SH.Account.1 ............................................................ 2169
SH.Athena.1 ............................................................. 2171
SH.AutoScaling.1 ....................................................... 2172
SH.AutoScaling.2 ....................................................... 2174
SH.AutoScaling.3 ....................................................... 2175
SH.AutoScaling.4 ....................................................... 2177
SH.AutoScaling.6 ....................................................... 2178
SH.AutoScaling.9 ....................................................... 2180
SH.Autoscaling.5 ....................................................... 2181
SH.CloudTrail.1 ........................................................ 2183
SH.CloudTrail.2 ........................................................ 2184
SH.CloudTrail.4 ........................................................ 2186
SH.CloudTrail.5 ........................................................ 2187
SH.CodeBuild.1 ......................................................... 2189
SH.CodeBuild.2 ......................................................... 2190
SH.CodeBuild.3 ......................................................... 2192
SH.CodeBuild.4 ......................................................... 2193
SH.CodeBuild.5 ......................................................... 2195
SH.DMS.1 ................................................................. 2196
SH.DynamoDB.1 ......................................................... 2198
SH.DynamoDB.2 ......................................................... 2199
SH.EC2.1 ................................................................. 2201
SH.EC2.10 ............................................................... 2202
SH.EC2.15 ............................................................... 2204
SH.EC2.16 ............................................................... 2205
| SH.IAM.21 | 2292 |
| SH.IAM.3 | 2294 |
| SH.IAM.4 | 2295 |
| SH.IAM.5 | 2297 |
| SH.IAM.6 | 2298 |
| SH.IAM.7 | 2300 |
| SH.IAM.8 | 2301 |
| SH.KMS.1 | 2303 |
| SH.KMS.2 | 2304 |
| SH.KMS.3 | 2306 |
| SH.Kinesis.1 | 2307 |
| SH.Lambda.1 | 2309 |
| SH.Lambda.2 | 2310 |
| SH.Lambda.5 | 2312 |
| SH.Neptune.1 | 2313 |
| SH.Neptune.2 | 2315 |
| SH.Neptune.3 | 2317 |
| SH.Neptune.4 | 2318 |
| SH.Neptune.5 | 2320 |
| SH.Neptune.6 | 2321 |
| SH.Neptune.7 | 2323 |
| SH.Neptune.8 | 2324 |
| SH.NetworkFirewall.3 | 2326 |
| SH.NetworkFirewall.4 | 2327 |
| SH.NetworkFirewall.5 | 2329 |
| SH.NetworkFirewall.6 | 2330 |
| SH.OpenSearch.1 | 2332 |
| SH.OpenSearch.2 | 2333 |
| SH.OpenSearch.3 | 2335 |
| SH.OpenSearch.4 | 2336 |
| SH.OpenSearch.5 | 2338 |
| SH.OpenSearch.6 | 2339 |
| SH.OpenSearch.7 | 2341 |
| SH.OpenSearch.8 | 2342 |
| SH.RDS.1 | 2344 |
| SH.RDS.10 | 2346 |
| SH.RDS.11 | 2347 |
| SH.RDS.13 | 2349 |
| SH.RDS.17 | 2350 |
| SH.RDS.18 | 2352 |
| SH.RDS.19 | 2353 |
| SH.RDS.2 | 2355 |
| SH.RDS.20 | 2357 |
| SH.RDS.21 | 2358 |
| SH.RDS.22 | 2360 |
| SH.RDS.23 | 2361 |
| SH.RDS.25 | 2363 |
| SH.RDS.27 | 2364 |
| SH.RDS.3 | 2366 |
| SH.RDS.4 | 2367 |
| SH.RDS.5 | 2369 |
| SH.RDS.6 | 2370 |
| SH.RDS.8 | 2372 |
| SH.RDS.9 | 2373 |
| SH.Redshift.1 | 2375 |
| SH.Redshift.10 | 2376 |
| SH.Redshift.2 | 2378 |
What Is AWS Control Tower?

AWS Control Tower offers a straightforward way to set up and govern an AWS multi-account environment, following prescriptive best practices. AWS Control Tower orchestrates the capabilities of several other AWS services, including AWS Organizations, AWS Service Catalog, and AWS IAM Identity Center, to build a landing zone in less than an hour. Resources are set up and managed on your behalf.

AWS Control Tower orchestration extends the capabilities of AWS Organizations. To help keep your organizations and accounts from drift, which is divergence from best practices, AWS Control Tower applies controls (sometimes called guardrails). For example, you can use controls to help ensure that security logs and necessary cross-account access permissions are created, and not altered.

If you are hosting more than a handful of accounts, it's beneficial to have an orchestration layer that facilitates account deployment and account governance. You can adopt AWS Control Tower as your primary way to provision accounts and infrastructure. With AWS Control Tower, you can more easily adhere to corporate standards, meet regulatory requirements, and follow best practices.

AWS Control Tower enables end users on your distributed teams to provision new AWS accounts quickly, by means of configurable account templates in Account Factory. Meanwhile, your central cloud administrators can monitor that all accounts are aligned with established, company-wide compliance policies.

In short, AWS Control Tower offers the easiest way to set up and govern a secure, compliant, multi-account AWS environment based on best practices established by working with thousands of enterprises. For more information about the working with AWS Control Tower and the best practices outlined in the AWS multi-account strategy, see AWS multi-account strategy: Best practices guidance (p. 36).

Features

AWS Control Tower has the following features:

- **Landing zone** – A landing zone is a well-architected, multi-account environment that's based on security and compliance best practices. It is the enterprise-wide container that holds all of your organizational units (OUs), accounts, users, and other resources that you want to be subject to compliance regulation. A landing zone can scale to fit the needs of an enterprise of any size.

- **Controls** – A control (sometimes called a guardrail) is a high-level rule that provides ongoing governance for your overall AWS environment. It's expressed in plain language. Three kinds of controls exist: preventive, detective, and proactive. Three categories of guidance apply to controls: mandatory, strongly recommended, or elective. For more information about controls, see How controls work (p. 9).

- **Account Factory** – An Account Factory is a configurable account template that helps to standardize the provisioning of new accounts with pre-approved account configurations. AWS Control Tower offers a built-in Account Factory that helps automate the account provisioning workflow in your organization. For more information, see Provision and manage accounts with Account Factory (p. 121).

- **Dashboard** – The dashboard offers continuous oversight of your landing zone to your team of central cloud administrators. Use the dashboard to see provisioned accounts across your enterprise, controls enabled for policy enforcement, controls enabled for continuous detection of policy non-conformance, and noncompliant resources organized by accounts and OUs.
How AWS Control Tower interacts with other AWS services

AWS Control Tower is built on top of trusted and reliable AWS services including AWS Service Catalog, AWS IAM Identity Center, and AWS Organizations. For more information, see Integrated services (p. 1377).

You can incorporate AWS Control Tower with other AWS services into a solution that helps you migrate your existing workloads to AWS. For more information, see How to take advantage of AWS Control Tower and CloudEndure to migrate workloads to AWS.

Configuration, Governance, and Extensibility

- **Automated account configuration**: AWS Control Tower automates account deployment and enrollment by means of an Account Factory (or “vending machine”), which is built as an abstraction on top of provisioned products in AWS Service Catalog. The Account Factory can create and enroll AWS accounts, and it automates the process of applying controls and policies to those accounts.

- **Centralized governance**: By employing the capabilities of AWS Organizations, AWS Control Tower sets up a framework that ensures consistent compliance and governance across your multi-account environment. The AWS Organizations service provides essential capabilities for managing a multi-account environment, including central governance and management of accounts, account creation from AWS Organizations APIs, and service control policies (SCPs).

- **Extensibility**: You can build or extend your own AWS Control Tower environment by working directly in AWS Organizations, as well as in the AWS Control Tower console. You can see your changes reflected in AWS Control Tower after you register your existing organizations and enroll your existing accounts into AWS Control Tower. You can update your AWS Control Tower landing zone to reflect your changes. If your workloads require further advanced capabilities, you can leverage other AWS partner solutions along with AWS Control Tower.

Are You a First-Time User of AWS Control Tower?

If you’re a first-time user of this service, we recommend that you read the following:

1. If you need more information about how to plan and organize your landing zone, see Plan your AWS Control Tower landing zone (p. 33) and AWS multi-account strategy for your AWS Control Tower landing zone (p. 35).
2. If you’re ready to create your first landing zone, see Getting started with AWS Control Tower (p. 16).
3. For information on drift detection and prevention, see Detect and resolve drift in AWS Control Tower (p. 169).
4. For security details, see Security in AWS Control Tower (p. 1386).
5. For information on updating your landing zone and member accounts, see Configuration update management in AWS Control Tower (p. 46).

How AWS Control Tower Works

This section describes at a high level how AWS Control Tower works. Your landing zone is a well-architected multi-account environment for all of your AWS resources. You can use this environment to enforce compliance regulations on all of your AWS accounts.
Structure of an AWS Control Tower Landing Zone

The structure of a landing zone in AWS Control Tower is as follows:

- **Root** – The parent that contains all other OUs in your landing zone.
- **Security OU** – This OU contains the Log Archive and Audit accounts. These accounts often are referred to as *shared accounts*. When you launch your landing zone, you can choose customized names for these shared accounts, and you have the option to bring existing AWS accounts into AWS Control Tower for security and logging. However, these cannot be renamed later, and existing accounts cannot be added for security and logging after initial launch.
- **Sandbox OU** – The Sandbox OU is created when you launch your landing zone, if you enable it. This and other registered OUs contain the enrolled accounts that your users work with to perform their AWS workloads.
- **IAM Identity Center directory** – This directory houses your IAM Identity Center users. It defines the scope of permissions for each IAM Identity Center user.
- **IAM Identity Center users** – These are the identities that your users can assume to perform their AWS workloads in your landing zone.

What happens when you set up a landing zone

When you set up a landing zone, AWS Control Tower performs the following actions in your management account on your behalf:

- Creates two AWS Organizations organizational units (OUs): Security, and Sandbox (optional), contained within the organizational root structure.
- Creates or adds two shared accounts in the Security OU: the Log Archive account and the Audit account.
- Creates a cloud-native directory in IAM Identity Center, with preconfigured groups and single sign-on access, if you choose the default AWS Control Tower configuration, or it allows you to self-manage your identity provider.
- Applies all mandatory, preventive controls to enforce policies.
- Applies all mandatory, detective controls to detect configuration violations.
- Preventive controls are not applied to the management account.
- Except for the management account, controls are applied to the organization as a whole.

Safely Managing Resources Within Your AWS Control Tower Landing Zone and Accounts

- When you create your landing zone, a number of AWS resources are created. To use AWS Control Tower, you must not modify or delete these AWS Control Tower managed resources outside of the supported methods described in this guide. Deleting or modifying these resources will cause your landing zone to enter an unknown state. For details, see [Guidance for creating and modifying AWS Control Tower resources](#).
- When you enable optional controls (those with *strongly recommended* or *elective* guidance), AWS Control Tower creates AWS resources that it manages in your accounts. Do not modify or delete resources created by AWS Control Tower. Doing so can result in the controls entering an unknown state. For more information, see [The AWS Control Tower controls library](#).

What Are the Shared Accounts?

In AWS Control Tower, the shared accounts in your landing zone are provisioned during setup: the management account, the log archive account, and the audit account.
What is the management account?

This is the account that you created specifically for your landing zone. This account is used for billing for everything in your landing zone. It’s also used for Account Factory provisioning of accounts, as well as to manage OUs and controls.

**Note**

It is not recommended to run any type of production workloads from an AWS Control Tower management account. Create a separate AWS Control Tower account to run your workloads.

When you set up your landing zone, the following AWS resources are created within your management account.

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Organizations</td>
<td>Accounts</td>
<td>audit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>log archive</td>
</tr>
<tr>
<td>AWS Organizations</td>
<td>OUs</td>
<td>Security</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sandbox</td>
</tr>
<tr>
<td>AWS Organizations</td>
<td>Service Control Policies</td>
<td>aws-guardrails-*</td>
</tr>
<tr>
<td>AWS CloudFormation</td>
<td>Stacks</td>
<td>AWSControlTowerBP-BASELINE-CLOUDTRAIL-MASTER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerBP-BASELINE-CONFIG-MASTER (in version 2.6 and later)</td>
</tr>
<tr>
<td>AWS CloudFormation</td>
<td>StackSets</td>
<td>AWSControlTowerBP-BASELINE-CLOUDTRAIL (Not deployed in 3.0 and later)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerBP_BASELINE_SERVICE_LINKED_ROLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerBP-BASELINE-CLOUDWATCH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerBP-BASELINE-CONFIG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerBP-BASELINE-ROLES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerBP-BASELINE-SERVICE-ROLES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerBP-BASELINE-SECURITY-TOPICS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-READ-PROHIBITED</td>
</tr>
</tbody>
</table>
### What Are the Shared Accounts?

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Control Tower Guardrail</td>
<td></td>
<td>AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-WRITE-PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerLoggingResources</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerSecurityResources</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerExecutionRole</td>
</tr>
<tr>
<td>AWS Service Catalog</td>
<td>Product</td>
<td>AWS Control Tower Account Factory</td>
</tr>
<tr>
<td>AWS Config</td>
<td>Aggregator</td>
<td>aws-controltower-ConfigAggregatorForOrganizations</td>
</tr>
<tr>
<td>AWS CloudTrail</td>
<td>Trail</td>
<td>aws-controltower-BaselineCloudTrail</td>
</tr>
<tr>
<td>Amazon CloudWatch</td>
<td>CloudWatch Logs</td>
<td>aws-controltower/CloudTrailLogs</td>
</tr>
<tr>
<td>AWS Identity and Access Management</td>
<td>Roles</td>
<td>AWSControlTowerAdmin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerStackSetRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerCloudTrailRolePolicy</td>
</tr>
<tr>
<td>AWS Identity and Access Management</td>
<td>Policies</td>
<td>AWSControlTowerServiceRolePolicy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerAdminPolicy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerCloudTrailRolePolicy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerStackSetRolePolicy</td>
</tr>
<tr>
<td>AWS IAM Identity Center</td>
<td>Directory groups</td>
<td>AWSAccountFactory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSAuditAccountAdmins</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerAdmins</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSLogArchiveAdmins</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSLogArchiveViewers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSSecurityAuditors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSSecurityAuditPowerUsers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSServiceCatalogAdmins</td>
</tr>
</tbody>
</table>
AWS Control Tower User Guide
What Are the Shared Accounts?

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS IAM Identity Center</td>
<td>Permission Sets</td>
<td>AWSAdministratorAccess</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSPowerUserAccess</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSServiceCatalogAdminFullAccess</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSServiceCatalogEndUserAccess</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSReadOnlyAccess</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSOrganizationsFullAccess</td>
</tr>
</tbody>
</table>

**Note**
The AWS CloudFormation StackSet BP_BASELINE_CLOUDTRAIL is not deployed in landing zone versions 3.0 or later. However, it continues to exist in earlier versions of the landing zone, until you update your landing zone.

**What is the log archive account?**

This account works as a repository for logs of API activities and resource configurations from all accounts in the landing zone.

When you set up your landing zone, the following AWS resources are created within your log archive account.

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS CloudFormation</td>
<td>Stacks</td>
<td>StackSet-AWSControlTowerGuardrailAWS-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GR-AUDIT-BUCKET-PUBLIC-READ-PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerGuardrailAWS-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GR-AUDIT-BUCKET-PUBLIC-WRITE-PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-CONFIG-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-CLOUDTRAIL-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-SERVICE-ROLES-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-SERVICE-LINKED-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ROLE-(In 3.2 and later)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-ROLES-</td>
</tr>
</tbody>
</table>
# What Are the Shared Accounts?

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Config</td>
<td>AWS Config Rules</td>
<td>AWSControlTower.AWS-GR_AUDIT_BUCKET_PUBLIC_READ_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTower.AWS-GR_AUDIT_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
</tr>
<tr>
<td>AWS CloudTrail</td>
<td>Trails</td>
<td>aws-controltower-BaselineCloudTrail</td>
</tr>
<tr>
<td>Amazon CloudWatch</td>
<td>CloudWatch Event Rules</td>
<td>aws-controltower-ConfigComplianceChangeEventRule</td>
</tr>
<tr>
<td>Amazon CloudWatch</td>
<td>CloudWatch Logs</td>
<td>aws-controltower/CloudTrailLogs /aws/lambda/aws-controltower-NotificationForwarder</td>
</tr>
<tr>
<td>AWS Identity and Access Management</td>
<td>Roles</td>
<td>aws-controltower-AdministratorExecutionRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-CloudWatchLogsRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-ConfigRecorderRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-ForwardSnsNotificationRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-ReadOnlyExecutionRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerExecution</td>
</tr>
<tr>
<td>AWS Identity and Access Management</td>
<td>Policies</td>
<td>AWSControlTowerServiceRolePolicy</td>
</tr>
<tr>
<td>Amazon Simple Notification Service</td>
<td>Topics</td>
<td>aws-controltower-SecurityNotifications</td>
</tr>
<tr>
<td>AWS Lambda</td>
<td>Applications</td>
<td>StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-*</td>
</tr>
<tr>
<td>AWS Lambda</td>
<td>Functions</td>
<td>aws-controltower-NotificationForwarder</td>
</tr>
<tr>
<td>Amazon Simple Storage Service</td>
<td>Buckets</td>
<td>aws-controltower-logs-*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-s3-access-logs-*</td>
</tr>
</tbody>
</table>
## What is the audit account?

The audit account is a restricted account that's designed to give your security and compliance teams read and write access to all accounts in your landing zone. From the audit account, you have programmatic access to review accounts, by means of a role that is granted to Lambda functions only. The audit account does not allow you to log in to other accounts manually. For more information about Lambda functions and roles, see [Configure a Lambda function to assume a role from another AWS account](#).

When you set up your landing zone, the following AWS resources are created within your audit account.

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS CloudFormation</td>
<td>Stacks</td>
<td>StackSet-AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-READ-PROHIBITED-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-WRITE-PROHIBITED-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-CONFIG-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-CLOUDTRAIL-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-SERVICE-ROLES-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-SERVICE-LINKED-ROLE-(In 3.2 and later)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-SECURITY-TOPICS-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-ROLES-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerSecurityResources-</td>
</tr>
<tr>
<td>AWS Config</td>
<td>Aggregator</td>
<td>aws-controltower-GuardrailsComplianceAggregator</td>
</tr>
<tr>
<td>AWS Config</td>
<td>AWS Config Rules</td>
<td>AWSControlTower.AWS-GR_AUDIT_BUCKET_PUBLIC_READ_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTower.AWS-GR_AUDIT_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
</tr>
<tr>
<td>AWS CloudTrail</td>
<td>Trail</td>
<td>aws-controltower-BaselineCloudTrail</td>
</tr>
</tbody>
</table>
How controls work

A control is a high-level rule that provides ongoing governance for your overall AWS environment. Each control enforces a single rule, and it's expressed in plain language. You can change the elective or strongly recommended controls that are in force, at any time, from the AWS Control Tower console or the AWS Control Tower APIs. Mandatory controls are always applied, and they can't be changed.

Preventive controls prevent actions from occurring. For example, the elective control called Disallow Changes to Bucket Policy for Amazon S3 Buckets (previously called Disallow Policy Changes to Log

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon CloudWatch</td>
<td>CloudWatch Event Rules</td>
<td>aws-controltower-ConfigComplianceChangeEventRule</td>
</tr>
<tr>
<td>Amazon CloudWatch</td>
<td>CloudWatch Logs</td>
<td>aws-controltower/CloudWatchLogs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/aws/lambda/aws-controltower-NotificationForwarder</td>
</tr>
<tr>
<td>AWS Identity and Access</td>
<td>Roles</td>
<td>aws-controltower-AdministratorExecutionRole</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td>aws-controltower-CloudWatchLogsRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-ConfigRecorderRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-ForwardSnsNotificationRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-ReadOnlyExecutionRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-AuditAdministratorRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-AuditReadOnlyRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerExecution</td>
</tr>
<tr>
<td>AWS Identity and Access</td>
<td>Policies</td>
<td>AWSControlTowerServiceRolePolicy</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amazon Simple Notification</td>
<td>Topics</td>
<td>aws-controltower-AggregateSecurityNotifications</td>
</tr>
<tr>
<td>Service</td>
<td></td>
<td>aws-controltower-AllConfigNotifications</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-SecurityNotifications</td>
</tr>
<tr>
<td>AWS Lambda</td>
<td>Functions</td>
<td>aws-controltower-NotificationForwarder</td>
</tr>
</tbody>
</table>
Archive) prevents any IAM policy changes within the log archive shared account. Any attempt to perform a prevented action is denied and logged in CloudTrail. The resource is also logged in AWS Config.

Detective controls detect specific events when they occur and log the action in CloudTrail. For example, the strongly recommended control called Detect Whether Encryption is Enabled for Amazon EBS Volumes Attached to Amazon EC2 Instances detects whether an unencrypted Amazon EBS volume is attached to an EC2 instance in your landing zone.

Proactive controls check whether resources are compliant with your company policies and objectives, before the resources are provisioned in your accounts. If the resources are out of compliance, they are not provisioned. Proactive controls monitor resources that would be deployed in your accounts by means of AWS CloudFormation templates.

For those who are familiar with AWS: In AWS Control Tower preventive controls are implemented with Service Control Policies (SCPs). Detective controls are implemented with AWS Config rules. Proactive controls are implemented with AWS CloudFormation hooks.

Related Topics

• About controls in AWS Control Tower (p. 196)
• Detect and resolve drift in AWS Control Tower (p. 169)

How AWS Control Tower Works With StackSets

AWS Control Tower uses AWS CloudFormation StackSets to set up resources in your accounts. Each stack set has StackInstances that correspond to accounts, and to AWS Regions per account. AWS Control Tower deploys one stack set instance per account and Region.

AWS Control Tower applies updates to certain accounts and AWS Regions selectively, based on AWS CloudFormation parameters. When updates are applied to some stack instances, other stack instances may be left in Outdated status. This behavior is expected and normal.

When a stack instance goes into Outdated status, it usually means that the stack corresponding to that stack instance is not aligned with the latest template in the stack set. The stack remains in the older template, so it might not include the latest resources or parameters. The stack is still completely usable.

Here's a quick summary of what behavior to expect, based on AWS CloudFormation parameters that are specified during an update:

If the stack set update includes changes to the template (that is, if the TemplateBody or TemplateURL properties are specified), or if the Parameters property is specified, AWS CloudFormation marks all stack instances with a status of Outdated prior to updating the stack instances in the specified accounts and AWS Regions. If the stack set update does not include changes to the template or parameters, AWS CloudFormation updates the stack instances in the specified accounts and Regions, while leaving all other stack instances with their existing stack instance status. To update all of the stack instances associated with a stack set, do not specify the Accounts or Regions properties.

For more information, see Update Your Stack Set in the AWS CloudFormation User Guide.
Terminology

Here’s a quick review of some terms you’ll see in the AWS Control Tower documentation.

First, it’s good to know that AWS Control Tower shares a lot of terminology with the AWS Organizations service, including the terms organization and organizational unit (OU), which appear throughout this document.

- For more information about organizations and OUs, see [AWS Organizations terminology and concepts](#). If you’re new to AWS Control Tower, that terminology is a good place to begin.
- **AWS Organizations** is an AWS service that helps you centrally govern your environment as you grow and scale your workloads on AWS. AWS Control Tower relies on AWS Organizations to create accounts, to enforce preventive controls at the OU level, and to provide centralized billing.
- An **AWS Account Factory account** is an AWS account provisioned using Account Factory in AWS Control Tower. Sometimes, Account Factory is referred to informally as a “vending machine” for accounts.
- Your AWS Control Tower **home Region** is the AWS Region in which your AWS Control Tower landing zone was deployed. You can view your home Region in your landing zone settings.
- **AWS Service Catalog** allows you to manage commonly deployed IT services, centrally. In the context of this document, Account Factory uses AWS Service Catalog to provision new AWS accounts, including accounts from customized blueprints.
- **AWS CloudFormation StackSets** are a type of resource that extends the functionality of stacks so that you can create, update, or delete stacks across multiple accounts and Regions with a single operation and a single CloudFormation template.
- A **stack instance** is a reference to a stack in a target account within a Region.
- A **stack** is a collection of AWS resources that you can manage as a single unit.
- An **aggregator** is an AWS Config resource type that collects AWS Config configuration and compliance data from multiple accounts and Regions within the organization, allowing you to view and query this compliance data within a single account.
- A **conformance pack** is a collection of AWS Config rules and remediation actions that can be deployed as a single entity in an account and a Region, or across an organization in AWS Organizations. You can use a conformance pack to help customize your AWS Control Tower environment. For technical blogs that provide more details, see [Related information](#).
- **Baseline**: To baseline an account is to set up its blueprints and controls. The baselining process also sets up the centralized logging and security audit roles on the account, as part of deploying the blueprints. AWS Control Tower baselines are contained in the roles that you apply to every enrolled account.
- **Blueprint**: A blueprint is an artifact that encapsulates some metadata, which describes infrastructure components that are deployed within an account. For example, an AWS CloudFormation template can serve as a blueprint for an AWS Control Tower account.
- **Drift**: A change in a resource installed by and configured by AWS Control Tower. Resources without drift enable AWS Control Tower to function properly.
- **Non-compliant resource**: A resource that is in violation of an AWS Config rule that defines a particular detective control.
- **Shared account**: One of the three accounts that AWS Control Tower creates automatically when you set up your landing zone: the management account, the log archive account, and the audit account. You can choose customized names for the log archive account and the audit account, during setup.
- **Member account**: A member account belongs to the AWS Control Tower organization. The member account can be **enrolled** or **unenrolled** in AWS Control Tower. When a registered OU contains a mix of enrolled and unenrolled accounts:
• Preventive controls enabled on the OU apply to all accounts within it, including unenrolled ones. This is true because preventive controls are enforced with SCPs at the OU level, not the account level. For more information, see Inheritance for service control policies in the AWS Organizations documentation.

• Detective controls enabled on the OU do not apply to unenrolled accounts.

An account can be a member of only one organization at a time, and its charges are billed to the management account for that organization. A member account can be moved to the root container of an organization.

• AWS account: An AWS account acts as a resource container and resource isolation boundary. An AWS account can be associated with billing and payment. An AWS account is different than a user account (sometimes called an IAM user account) in AWS Control Tower. Accounts created through the Account Factory provisioning process are AWS accounts. AWS accounts also can be added to AWS Control Tower by means of the account enrollment or OU registration process.

• Control: A control (also known as a guardrail) is a high-level rule that provides ongoing governance for your overall AWS Control Tower environment. Each control enforces a single rule. Preventive controls are implemented with SCPs. Detective controls are implemented with AWS Config rules. Proactive controls are implemented with AWS CloudFormation hooks. For more information, see How controls work (p. 9).

• Landing zone: A landing zone is a cloud environment that offers a recommended starting point, including default accounts, account structure, network and security layouts, and so forth. From a landing zone, you can deploy workloads that utilize your solutions and applications.

• Nested OU: A nested OU in AWS Control Tower is an OU contained within another OU. A nested OU can have exactly one parent OU, and each account can be a member of exactly one OU. Nested OUs create a hierarchy. When you attach a policy to one of the OUs in the hierarchy, it flows down and affects all the OUs and accounts beneath it. A nested OU hierarchy in AWS Control Tower can be a maximum of five levels deep.

• Parent OU: The OU immediately above the current OU in the hierarchy. Each OU can have exactly one parent OU.

• Child OU: Any OU below the current OU in the hierarchy. An OU can have many child OUs.

• OU hierarchy: In AWS Control Tower, the hierarchy of nested OUs can have up to five levels. The order of nesting is referred to as Levels. The top of the hierarchy is designated as Level 1.

• Top-level OU: A top-level OU is any OU that's directly under the Root, not the Root itself. The Root is not considered an OU.
Pricing

No additional charge exists for using AWS Control Tower. You only pay for the AWS services enabled by AWS Control Tower, and the services you use in your landing zone. For example, you pay for Service Catalog for provisioning accounts with Account Factory, and AWS CloudTrail for events tracked in your landing zone. For information about the pricing and fees associated with AWS Control Tower, see AWS Control Tower pricing.

If you are running ephemeral workloads from accounts in AWS Control Tower, you may see an increase in costs associated with AWS Config. For details, see AWS Config pricing. Contact your AWS account representative for more specific information about managing these costs. To learn more about how AWS Config works with AWS Control Tower, see Monitoring resource changes with AWS Config (p. 1413).

If you implement AWS CloudTrail trails outside of AWS Control Tower, you can use them with AWS Control Tower. However, you may incur duplicate charges, if you also opt in to trails managed by AWS Control Tower. We do not recommend setting up external trails, unless you have a specific requirement. If you choose to opt in during landing zone setup or update, AWS Control Tower sets up and activates an organization-level CloudTrail trail for you in the management account. For information about managing CloudTrail costs, see Managing CloudTrail costs.
Setting up

Before you use AWS Control Tower for the first time, follow the steps in this section to create an AWS account and protect your AWS Control Tower management account. For information on additional setup tasks specifically for AWS Control Tower, see Getting started with AWS Control Tower (p. 16).

Sign up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including AWS Control Tower. If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

Note your AWS account number, because you need it for other tasks.

Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

   When you sign up for an AWS account, an AWS account root user is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to an administrative user, and use only the root user to perform tasks that require root user access.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing My Account.

Create an administrative user

After you sign up for an AWS account, create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the AWS Management Console as the account owner by choosing Root user and entering your AWS account email address. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the AWS Sign-In User Guide.

2. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the IAM User Guide.
Create an administrative user

- For your daily administrative tasks, grant administrative access to an administrative user in AWS IAM Identity Center.

  For instructions, see Getting started in the AWS IAM Identity Center User Guide.

Sign in as the administrative user

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

  For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the AWS Sign-In User Guide.

Security for your accounts

You can find additional guidance about how to set up best practices that protect the security of your AWS Control Tower accounts, in the AWS Organizations documentation.

- Best practices for the management account
- Best practices for member accounts

Next step

Getting started with AWS Control Tower (p. 16)
Getting started with AWS Control Tower

This getting started procedure is intended for AWS Control Tower administrators. Follow this procedure when you're ready to set up your landing zone. From start to finish, it should take about half an hour. This procedure requires some prerequisites and three main steps.

If you are an AWS customer currently, but new to AWS Control Tower, you may wish to review the section called Plan your AWS Control Tower landing zone (p. 33), before you proceed.

Topics

- AWS Control Tower quick start guide (p. 16)
- Prerequisite: Automated pre-launch checks for your management account (p. 17)
- Step 1: Create your shared account email addresses (p. 18)
- Expectations for landing zone configuration (p. 19)
- Step 2. Configure and launch your landing zone (p. 20)
- Step 3. Review and set up the landing zone (p. 25)
- Next steps (p. 26)

AWS Control Tower quick start guide

If you are new to AWS, you can follow the steps in this section to get started quickly with AWS Control Tower. If you prefer to customize your AWS Control Tower environment right away, see Step 2. Configure and launch your landing zone (p. 20).

Note
AWS Control Tower sets up paid services, such as AWS CloudTrail, AWS Config, Amazon CloudWatch, Amazon S3, and Amazon VPC. When used, these services may incur costs, as shown on the pricing page. The AWS management console shows you the usage of any paid services and the costs incurred. No additional costs are created by AWS Control Tower itself.

Before you begin

The most important decision to make before you begin the setup process is to choose your home Region. Your home Region is the AWS Region in which you'll run most of your workloads or store most of your data. It cannot be changed after you've set up your AWS Control Tower landing zone. For more information about how to choose a home Region, see Administrative tips for landing zone setup (p. 39).

Note
By default, AWS Control Tower chooses the Region in which your account is operating currently as your home Region. You can see your current Region in the upper right of your AWS management console screen.

The quick start procedure assumes that you'll accept the default values for the resources in your AWS Control Tower environment. Many of these choices can be changed later. A few one-time choices are listed in the section called Expectations for landing zone configuration (p. 19).

If you've created a new AWS account, it automatically meets the required prerequisites for setting up AWS Control Tower. You can proceed through the steps that follow.
Quick start steps

1. Sign in to the AWS management console with your administrator user credentials.
3. Verify that you are working in your desired home Region.
4. Choose Set up landing zone.
5. Follow the instructions in the console, accepting all the default values. You will need to type in the email address for your account, a log archive account, and an audit account.
6. Confirm your choices and choose Set up landing zone.
7. AWS Control Tower takes about 30 minutes to set up all of the resources in your landing zone.

For a more detailed version of how to set up AWS Control Tower, including ways to customize your environment, read and follow the procedures in the next few topics.

Note
If you are a first-time customer and you encounter a setup issue, contact AWS Support for diagnostic assistance.

Prerequisite: Automated pre-launch checks for your management account

Before AWS Control Tower sets up the landing zone, it automatically runs a series of pre-launch checks in your account. There's no action required on your part for these checks, which ensure that your management account is ready for the changes that establish your landing zone. Here are the checks that AWS Control Tower runs before setting up a landing zone:

- The existing service limits for the AWS account must be sufficient for AWS Control Tower to launch. For more information, see Limitations and quotas in AWS Control Tower (p. 27).
- The AWS account must be subscribed to the following AWS services:
  - Amazon Simple Storage Service (Amazon S3)
  - Amazon Elastic Compute Cloud (Amazon EC2)
  - Amazon SNS
  - Amazon Virtual Private Cloud (Amazon VPC)
  - AWS CloudFormation
  - AWS CloudTrail
  - Amazon CloudWatch
  - AWS Config
  - AWS Identity and Access Management (IAM)
  - AWS Lambda

  Note
  By default, all accounts are subscribed to these services.

Considerations for AWS IAM Identity Center (IAM Identity Center) customers

- If AWS IAM Identity Center (IAM Identity Center) is already set up, the AWS Control Tower home Region must be the same as the IAM Identity Center Region.
Step 1: Create your shared account email addresses

If you're setting up your landing zone in a new AWS account, see Setting up (p. 14).

To set up your landing zone with new shared accounts, AWS Control Tower requires two unique email addresses that aren't already associated with an AWS account. Each of these email addresses will serve...
as a collaborative inbox -- a shared email account -- intended for the various users in your enterprise that will do specific work related to AWS Control Tower.

- If you are setting up AWS Control Tower for the first time, and if you are bringing existing security and log archive accounts into AWS Control Tower, you can enter the current email addresses of the existing AWS accounts.

The email addresses are required for:

- **Audit account** – This account is for your team of users that need access to the audit information made available by AWS Control Tower. You can also use this account as the access point for third-party tools that will perform programmatic auditing of your environment to help you audit for compliance purposes.

- **Log archive account** – This account is for your team of users that need access to all the logging information for all of your enrolled accounts within registered OUs in your landing zone.

These accounts are set up in the **Security OU** when you create your landing zone. As a best practice, we recommend that when you perform actions in these accounts, you should use an IAM Identity Center user with the appropriately scoped permissions.

**Note**

If you specify existing AWS accounts as your **audit** and **log archive** accounts, the existing accounts must pass some pre-launch checks to ensure that no resources are in conflict with AWS Control Tower requirements. If these checks are not successful, your landing zone setup may not succeed. In particular, the accounts must not have existing AWS Config resources. For more information, see [Considerations for bringing existing security or logging accounts](p. 106).

For the sake of clarity, this User Guide always refers to the shared accounts by their default names: **log archive** and **audit**. As you read this document, remember to substitute the customized names you give to these accounts initially, if you choose to customize them. You can view your accounts with their customized names on the **Account details** page.

**Note**

We are changing our terminology regarding the default names of some AWS Control Tower organizational units (OUs) to align with the AWS multi-account strategy. You may notice some inconsistencies while we are making a transition to improve the clarity of these names. The Security OU was formerly called the Core OU. The Sandbox OU was formerly called the Custom OU.

## Expectations for landing zone configuration

The process of setting up your AWS Control Tower landing zone has multiple steps. Certain aspects of your AWS Control Tower landing zone are configurable. Other choices cannot be changed after setup.

### Key items to configure during setup

- You can select your top-level OU names during setup, and you also can change OU names after you’ve set up your landing zone. By default, the top-level OUs are named **Security** and **Sandbox**. For more information, see [Guidelines to set up a well-architected environment](p. 36).

- During setup, you can select customized names for the shared accounts that AWS Control Tower creates, called **log archive** and **audit** by default, but you cannot change these names after setup. (This is a one-time selection.)

- During setup, you can optionally specify existing AWS accounts for AWS Control Tower to use as audit and log archive accounts. If you plan to specify existing AWS accounts, and if those accounts have existing AWS Config resources, you must delete the existing AWS Config resources before you can enroll the accounts into AWS Control Tower. (This is a one-time selection.)
Step 2. Configure and launch your landing zone

Before you launch your AWS Control Tower landing zone, determine the most appropriate home Region. For more information, see Administrative tips for landing zone setup (p. 39).

Important

Changing your home Region after you have deployed your AWS Control Tower landing zone requires decommissioning as well as the assistance of AWS Support. This practice is not recommended.

AWS Control Tower has no APIs or programmatic access for launching your landing zone. To configure and launch your landing zone in the console, perform the following series of steps.

Prepare: Navigate to the AWS Control Tower console

2. In the console, verify that you are working in your desired home Region for AWS Control Tower. Then choose Set up your landing zone.

Step 2a. Review and select your AWS Regions

Be sure you've correctly designated the AWS Region that you select for your home Region. After you've deployed AWS Control Tower, you can't change the home Region.

In this section of the setup process, you can add any additional AWS Regions that you require. You can add more Regions at a later time, if needed, and you can remove Regions from governance.

To select additional AWS Regions to govern

1. The panel shows you the current Region selections. Open the dropdown menu to see a list of additional Regions available for governance.
2. Check the box next to each Region to bring into governance by AWS Control Tower. Your home Region selection is not editable.

To deny access to certain Regions
To deny access to AWS resources and workloads in certain AWS Regions, select **Enabled** in the section for the Region deny control. By default, the setting for this control is **Not enabled**.

### Step 2b. Configure your organizational units (OUs)

If you accept the default names of these OUs, there’s no action you need to take for setup to continue. To change the names of the OUs, enter the new names directly in the form field.

- **Foundational OU** – AWS Control Tower relies upon a Foundational OU that is initially named the **Security OU**. You can change the name of this OU during initial setup and afterward, from the OU details page. This Security OU contains your two shared accounts, which by default are called the **log archive** account and the **audit** account.

- **Additional OU** – AWS Control Tower can set up one or more Additional OUs for you. We recommend that you provision at least one Additional OU in your landing zone, besides the Security OU. If this Additional OU is intended for development projects, we recommend that you name it the **Sandbox OU**, as given in the [Guidelines to set up a well-architected environment](#) (p. 36). If you already have an existing OU in AWS Organizations, you may see the option to skip setting up an Additional OU in AWS Control Tower.

### Step 2c. Configure your shared accounts, logging, and encryption

In this section of the setup process, the panel shows the default selections for the names of your shared AWS Control Tower accounts. These accounts are an essential part of your landing zone. **Do not move or delete these shared accounts.** You can choose customized names for the **audit** and **log archive** accounts during setup. Alternatively, you have a one-time option to specify existing AWS accounts as your shared accounts.

You must provide unique email addresses for your log archive and audit accounts, and you can verify the email address that you previously provided for your management account. Choose the **Edit** button to change the editable default values.

**About the shared accounts**

- **The management account** – The AWS Control Tower management account is part of the Root level. The management account allows for AWS Control Tower billing. The account also has administrator permissions for your landing zone. You cannot create separate accounts for billing and for administrator permissions in AWS Control Tower.

  The email address shown for the management account is not editable during this phase of setup. It is shown as a confirmation, so you can check that you’re editing the correct management account, in case you have multiple accounts.

- **The two shared accounts** – You can choose customized names for these two accounts, or bring your own accounts, and you must supply a unique email address for each account, either new or existing. If you choose to have AWS Control Tower create new shared accounts for you, the email addresses must not already have associated AWS accounts.

**To configure the shared accounts, fill in the requested information.**

1. At the console, enter a name for the account initially called the **log archive** account. Many customers decide to keep the default name for this account.
2. Provide a unique email address for this account.
3. Enter a name for the account initially called the **audit** account. Many customers choose to call it the **Security** account.
4. Provide a unique email address for this account.

**Optionally configure log retention**

During this phase of setup, you can customize the log retention policy for Amazon S3 buckets that store your AWS CloudTrail logs in AWS Control Tower, in increments of days or years, up to a maximum of 15 years. If you choose not to customize your log retention, the default settings are one year for standard account logging and 10 years for access logging. This feature also is available when you update or repair your landing zone.

**Optionally self-manage AWS account access**

You can select whether AWS Control Tower sets up AWS account access with AWS Identity and Access Management (IAM), or whether to self-manage AWS account access—either with AWS IAM Identity Center users, roles, and permissions that you can set up and customize on your own, or with another method such as an external IdP, either for direct account federation or federation to multiple accounts by means of IAM Identity Center. You can change this selection later.

By default, AWS Control Tower sets up AWS IAM Identity Center for your landing zone, in alignment with best-practices guidance defined in Organizing your AWS environment using multiple accounts. Most customers choose the default. Alternative access methods are required sometimes, for regulatory compliance in specific industries or countries, or in AWS Regions where AWS IAM Identity Center is not available.

Selection of identity providers at the account level is not supported. This option applies only for the landing zone as a whole.

For more information, see IAM Identity Center guidance (p. 43).

**Optionally configure AWS CloudTrail trails**

As a best practice, we recommend that you set up logging. If you wish to allow AWS Control Tower to set up an organization-level CloudTrail trail and manage it for you, choose **Opt in**. If you wish to manage logging with your own CloudTrail trails or a third-party logging tool, choose **Opt out**. Confirm your selection when requested to do so in the console. You can change your selection, and opt into, or opt out of, organization-level trails when you update your landing zone.

You can set up and manage your own CloudTrail trails at any time, including organization-level and account-level trails. If you set up duplicate CloudTrail trails, you may incur duplicate costs when CloudTrail events are logged.

**Optionally configure AWS KMS keys**

If you wish to encrypt and decrypt your resources with an AWS KMS encryption key, select the checkbox. If you have existing keys, you'll be able to select them from identifiers displayed in a dropdown menu. You can generate a new key by choosing **Create a key**. You can add or change a KMS key any time you update your landing zone.

When you select **Set up landing zone**, AWS Control Tower performs a pre-check to validate your KMS key. The key must meet these requirements:

- Enabled
- Symmetric
- Not a multi-Region key
- Has correct permissions added to the policy
- Key is in the management account
You may see an error banner if the key does not meet these requirements. In that case, choose another key or generate a key. Be sure to edit the key’s permissions policy, as described in the next section.

**Update the KMS key policy**

Before you can update a KMS key policy, you must create a KMS key. For more information, see [Creating a key policy](#) in the AWS Key Management Service Developer Guide.

To use a KMS key with AWS Control Tower, you must update the default KMS key policy by adding the minimum required permissions for AWS Config and AWS CloudTrail. As a best practice, we recommend that you include the minimum required permissions in any policy. When updating a KMS key policy, you can add permissions as a group in a single JSON statement or line by line.

The procedure describes how to update the default KMS key policy in the AWS KMS console by adding policy statements that allow AWS Config and CloudTrail to use AWS KMS for encryption. The policy statements require that you include the following information:

- **YOUR-MANAGEMENT-ACCOUNT-ID** – the ID of the management account in which AWS Control Tower will be set up.
- **YOUR-HOME-REGION** – the home Region that you will select when setting up AWS Control Tower.
- **YOUR-KMS-KEY-ID** – the KMS key ID that will be used with the policy.

**To update the KMS key policy**

2. From the navigation pane, choose Customer managed keys.
3. In the table, select the key that you want to edit.
4. In the Key policy tab, make sure that you can view the key policy. If you can't view the key policy, choose Switch to policy view.
5. Choose Edit, and update the default KMS key policy by adding the following policy statements for AWS Config and CloudTrail.

**AWS Config policy statement**

```json
{
   "Sid": "Allow Config to use KMS for encryption",
   "Effect": "Allow",
   "Principal": {
      "Service": "config.amazonaws.com"
   },
   "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey"
   ],
}
```

**CloudTrail policy statement**

```json
{
   "Sid": "Allow CloudTrail to use KMS for encryption",
   "Effect": "Allow",
   "Principal": {
      "Service": "cloudtrail.amazonaws.com"
   },
   "Action": [
      "kms:GenerateDataKey"
   ]
}
```
Step 2c. Configure your shared accounts, logging, and encryption

The following example policy shows what your KMS key policy might look like after you add the policy statements that grant AWS Config and CloudTrail the minimum required permissions. The example policy doesn’t include your default KMS key policy.

```json
{
  "Version": "2012-10-17",
  "Id": "CustomKMSPolicy",
  "Statement": [
    {
      "Sid": "Allow Config to use KMS for encryption",
      "Effect": "Allow",
      "Principal": {
        "Service": "config.amazonaws.com"
      },
      "Action": ["kms:Decrypt", "kms:GenerateDataKey"],
    },
    {
      "Sid": "Allow CloudTrail to use KMS for encryption",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": ["kms:GenerateDataKey*", "kms:Decrypt"],
      "Condition": {
        "StringEquals": {
        },
        "StringLike": {
        }
      }
    }
  ]
}
```

6. Choose Save changes.

Example KMS key policy

The following example policy shows what your KMS key policy might look like after you add the policy statements that grant AWS Config and CloudTrail the minimum required permissions. The example policy doesn’t include your default KMS key policy.
To view other example policies, see the following pages:

- [Required Permissions for the KMS Key When Using Service-Linked Roles](https://docs.aws.amazon.com/config/latest/developerguide/config-servicelinked-roles.html) in the AWS Config Developer Guide.

**Protect against attackers**

For more information about how to help protect against attackers when you grant permissions to other AWS service principals, see [Optional conditions for your role trust relationships](https://docs.aws.amazon.com/iam/latest/userguide/trustpolicy-conditions.html) (p. 88). By adding certain conditions to your policies, you can help prevent a specific type of attack, known as a confused deputy attack, which occurs if an entity coerces a more-privileged entity to perform an action, such as with cross-service impersonation. For general information about policy conditions, also see [Specifying conditions in a policy](https://docs.aws.amazon.com/iam/latest/userguide/policiespecifying-conditions.html) (p. 1396).

The AWS Key Management Service (AWS KMS) allows you to create multi-Region KMS keys and asymmetric keys; however, AWS Control Tower does not support multi-Region keys or asymmetric keys. AWS Control Tower performs a pre-check of your existing keys. You may see an error message if you select a multi-Region key or an asymmetric key. In that case, generate another key for use with AWS Control Tower resources.

For more information about AWS KMS, see the [AWS KMS Developer Guide](https://docs.aws.amazon.com/kms/latest/userguide/what-is-kms.html).

Note that customer data in AWS Control Tower is encrypted at rest, by default, using SSE-S3.

**Optionally configure and create customized member accounts**

When you follow the [Create account](https://docs.aws.amazon.com/ControlTower/latest/userguide/create-account.html) workflow to add your member accounts, you can optionally specify a previously-defined blueprint to use for provisioning customized member accounts from the AWS Control Tower console. You can customize accounts later if you do not have a blueprint available. See [Customize accounts with Account Factory Customization (AFC)](https://docs.aws.amazon.com/ControlTower/latest/userguide/extend-your-control-tower.html) (p. 129).

---

**Step 3. Review and set up the landing zone**

The next section in the setup shows you the permissions that AWS Control Tower requires for your landing zone. Choose a checkbox to expand each topic. You'll be asked to agree to these permissions, which may affect multiple accounts, and to agree to the overall Terms of Service.

**To finalize**

1. At the console, review the Service permissions, and when you're ready, choose I understand the permissions AWS Control Tower will use to administer AWS resources and enforce rules on my behalf.
2. To finalize your selections and initialize launch, choose Set up landing zone.

This series of steps starts the process of setting up your landing zone, which can take about thirty minutes to complete. During setup, AWS Control Tower creates your Root level, the Security OU, and the shared accounts. Other AWS resources are created, modified, or deleted.
Confirm SNS subscriptions
The email address you provided for the audit account will receive AWS Notification – Subscription Confirmation emails from every AWS Region supported by AWS Control Tower. To receive compliance emails in your audit account, you must choose the Confirm subscription link within each email from each AWS Region supported by AWS Control Tower.

Next steps
Now that your landing zone is set up, it's ready for use.

To learn more about how you can use AWS Control Tower, see the following topics:

- For recommended administrative practices, see Best Practices.
- You can set up IAM Identity Center users and groups with specific roles and permissions. For recommendations, see Recommendations for setting up groups, roles, and policies (p. 40).
- To begin enrolling organizations and accounts from your AWS Organizations deployments, see Govern existing organizations and accounts.
- Your end users can provision their own AWS accounts in your landing zone using Account Factory. For more information, see Permissions for configuring and provisioning accounts (p. 121).
- To assure Compliance Validation for AWS Control Tower (p. 1407), your central cloud administrators can review log archives in the Log Archive account, and designated third-party auditors can review audit information in the Audit (shared) account, which is a member of the Security OU.
- To learn more about the capabilities of AWS Control Tower, see Related information.
- Try visiting a curated list of YouTube videos that explain more about how to use AWS Control Tower functionality.
- From time to time, you may need to update your landing zone to get the latest backend updates, the latest controls, and to keep your landing zone up-to-date. For more information, see Configuration update management in AWS Control Tower (p. 46).
- If you encounter issues while using AWS Control Tower, see Troubleshooting (p. 1445).

Important
If you have not yet enabled MFA for your account's root user, do so now. For more information about best practices for the root user, see Best practices to protect your account's root user.
Limitations and quotas in AWS Control Tower

This chapter covers the AWS service limitations and quotas that you should keep in mind as you use AWS Control Tower. If you’re unable to set up your landing zone due to a service quota issue, contact AWS Support.

For more information about limitations that are specific to controls, see Control limitations (p. 28).

Limitations in AWS Control Tower

This section describes known limitations and unsupported use cases in AWS Control Tower.

- AWS Control Tower has overall concurrency limitations. In general, one operation at a time is permitted. Two exceptions to this limitation are allowed:
  - Optional controls can be activated and deactivated concurrently, through an asynchronous process. Up to ten (10) control-related operations at a time can be in progress, in total, no matter if they are called from the console or from an API.
  - Accounts can be provisioned, updated, and enrolled concurrently in Account Factory, through an asynchronous process, with up to five (5) account-related operations in progress simultaneously. Unmanaging accounts must be performed one account at a time.
- Email addresses of shared accounts in the Security OU can be changed, but you must update your landing zone to see these changes in the AWS Control Tower console.
- A limit of five (5) SCPs per OU applies to OUs in your AWS Control Tower landing zone.
- AWS Control Tower supports up to 10,000 accounts in your landing zone’s organization, divided among all of your OUs.
- Existing OUs with over 300 directly nested accounts cannot be registered or re-registered in AWS Control Tower. For more information about limitations with registering OUs, see Regions and stack set limitations (p. 31).
- CfCT is unavailable in these AWS Regions, because some dependencies are not available:
  - Asia Pacific (Jakarta and Osaka)
  - Israel (Tel Aviv)
  - Middle East (UAE)
  - Europe (Spain)
  - Asia Pacific (Hyderabad)
  - Europe (Zurich)
- You can deploy resources to these Regions with CfCT, but you cannot build CfCT in these Regions.
- AFT is not available in the following AWS Regions, because some dependencies are not available:
  - Israel (Tel Aviv)
  - Middle East (UAE)
  - Europe (Spain)
  - Asia Pacific (Hyderabad)
  - Europe (Zurich)
- When calling a control API to activate or deactivate a control, the limit for EnableControl and DisableControl updates in AWS Control Tower is ten (10) concurrent operations. You may need to adjust your code to wait for completions.
When you are provisioning accounts with AFC, with blueprints that are based in Terraform, you can deploy those blueprints to only one AWS Region. By default, AWS Control Tower deploys to the home Region.

You can contact AWS Support to request a limit increase for some resources in AWS Control Tower. For example, you can request a limit increase from five of up to ten concurrent account-related operations. Some AWS Control Tower performance characteristics may change after a limit increase. For example, it may take longer to update an OU when you have more accounts in it. Or it may take longer to complete an action on OU with five SCPs than with three SCPs.

Video: Automate requests for service limit increase

This video (7:24) describes how to automate service limit increases for deployments in AWS Control Tower. It also shows how to automate enrollment of new accounts into AWS Enterprise support for your organization. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Managing Nested OUs in AWS Control Tower.

When provisioning new accounts in this environment, you can use lifecycle events to trigger automated requests for service limit increases in specified AWS Regions.

More information about AWS quotas is available in the AWS General Reference.

Control behavior also is limited in case of mixed governance. For more information, see Avoid mixed governance when configuring Regions (p. 99).

Control limitations

If you modify AWS Control Tower resources, such as an SCP, or remove any AWS Config resource, such as a Config recorder or aggregator, AWS Control Tower can no longer guarantee that the controls are functioning as designed. Therefore, the security of your multi-account environment may be compromised. The AWS shared responsibility model of security is applicable to any such changes you may make.

Note

AWS Control Tower helps maintain the integrity of your environment by resetting the SCPs of the controls to their standard configuration when you update your landing zone. Changes that you may have made to SCPs are replaced by the standard version of the control, by design.

Some controls in AWS Control Tower do not operate in certain AWS Regions where AWS Control Tower is available, because those Regions do not support the required underlying functionality. This limitation affects certain detective controls, certain proactive controls, and certain controls in the Security Hub Service-managed Standard: AWS Control Tower. For more information about Regional availability, see the Regional services list documentation and the Security Hub controls reference documentation.

For more information about how AWS Control Tower manages the limitations of Regions and controls, see Considerations for activating AWS opt-in Regions (p. 100).

You can view the Regions for each control in the AWS Control Tower console.

The following AWS Regions do not support controls that are part of the Security Hub Service-managed Standard: AWS Control Tower.

• Asia Pacific (Hong Kong) Region, ap-east-1
• Asia Pacific (Jakarta) Region, ap-southeast-3
• Asia Pacific (Osaka) Region, ap-northeast-3
Control limitations

- Europe (Milan) Region, eu-south-1
- Africa (Cape Town) Region, af-south-1
- Middle East (Bahrain) Region, me-south-1
- Israel (Tel Aviv), il-central-1
- Middle East (UAE) Region, me-central-1
- Europe (Spain) Region, eu-south-2
- Asia Pacific (Hyderabad) Region, ap-south-2
- Europe (Zurich) Region, eu-central-2
- Asia Pacific (Melbourne) Region, ap-southeast-4

The following Regions do not support IAM Identity Center.

- Middle East (UAE) Region, me-central-1
- Asia Pacific (Hyderabad) Region, ap-south-2

For more information about AWS Regions and support for IAM Identity Center, see Regions and endpoints in the AWS Identity and Access Management User Guide.

The following Regions do not support CfCT, due to a dependency on AWS CodePipeline, which is not available in these Regions.

- Asia Pacific (Jakarta) Region, ap-southeast-3
- Asia Pacific (Osaka) Region, ap-northeast-3
- Asia Pacific (Melbourne) Region, ap-southeast-4

You can deploy and manage resources in these Regions with CfCT, if you deploy CfCT to your AWS Control Tower home Region, but you cannot build CfCT in these Regions.

The following table shows proactive controls that are not supported in certain AWS Regions.

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Unsupported regions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.REDSHIFT.PR.5</td>
<td>ap-southeast-4, ap-south-2, ap-southeast-3, eu-central-2, eu-south-2, il-central-1, me-central-1</td>
</tr>
<tr>
<td>CT.GLUE.PR.2</td>
<td>Unsupported</td>
</tr>
<tr>
<td>CT.DAX.PR.2</td>
<td>us-west-1</td>
</tr>
</tbody>
</table>

The following table shows detective controls that are not supported in certain AWS Regions.

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Unsupported regions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-GR_AUTOSCALING_LAUNCH_CONFIG_PUBLIC_IP_DISABLED</td>
<td>ap-northeast-3, ap-southeast-3, il-central-1, ap-enabled-4</td>
</tr>
<tr>
<td>AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
<td>ap-northeast-3, ap-southeast-3</td>
</tr>
<tr>
<td>AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
<td>ap-northeast-3, eu-south-2</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Unsupported regions</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>AWS-GR_EMR_MASTER_NO_PUBLIC_IP</td>
<td>ap-northeast-3, ap-southeast-3, af-south-1, eu-south-1, il-central-1, me-central-1, eu-south-2, ap-south-2, eu-central-2, ap-southeast-4</td>
</tr>
<tr>
<td>AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK</td>
<td>ap-northeast-3, ap-southeast-3, eu-south-2</td>
</tr>
<tr>
<td>AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td>ap-northeast-3, ap-southeast-3, ap-south-2, eu-south-2</td>
</tr>
<tr>
<td>AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS</td>
<td>ap-northeast-3, ap-southeast-3, af-south-1, eu-south-1, il-central-1, me-central-1, eu-south-2, ap-south-2, eu-central-2, ap-southeast-4</td>
</tr>
<tr>
<td>AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
<td>ap-northeast-3</td>
</tr>
<tr>
<td>AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS</td>
<td>ap-northeast-3, ap-southeast-3, af-south-1, eu-south-1, us-west-1, il-central-1, me-central-1, eu-south-2, ap-south-2, eu-central-2, ap-southeast-4</td>
</tr>
<tr>
<td>AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
<td>ap-northeast-3, ap-southeast-3, il-central-1, eu-south-2, ap-south-2, eu-central-2, ap-southeast-4</td>
</tr>
<tr>
<td>AWS-GR_RESTRICTED_SSH</td>
<td>ap-northeast-3, ap-southeast-3, af-south-1, eu-south-1, il-central-1, me-central-1, eu-south-2, ap-south-2, eu-central-2, ap-southeast-4</td>
</tr>
<tr>
<td>AWS-GR_DMS_REPLICATION_NOT_PUBLIC</td>
<td>ap-northeast-3, ap-southeast-3, af-south-1, eu-south-1, il-central-1, me-central-1, eu-south-2, ap-south-2, eu-central-2, ap-southeast-4</td>
</tr>
<tr>
<td>AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
<td>ap-northeast-3, ap-southeast-3, af-south-1, eu-south-1, il-central-1, eu-south-2, eu-central-2, ap-southeast-4</td>
</tr>
<tr>
<td>AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
<td>ap-northeast-3</td>
</tr>
<tr>
<td>AWS-GR_ENCRYPTED_VOLUMES</td>
<td>ap-northeast-3, af-south-1, eu-south-1, il-central-1</td>
</tr>
<tr>
<td>AWS-GR_RESTRICTED_COMMON_PORTS</td>
<td>ap-northeast-3, ap-southeast-3, af-south-1, eu-south-1, il-central-1, me-central-1, eu-south-2, eu-central-2</td>
</tr>
<tr>
<td>AWS-GR_IAM_USER_MFA_ENABLED</td>
<td>il-central-1, me-central-1, eu-south-2, ap-south-2, eu-central-2, ap-southeast-4</td>
</tr>
<tr>
<td>AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS</td>
<td>il-central-1, me-central-1, eu-south-2, ap-south-2, eu-central-2, ap-southeast-4</td>
</tr>
<tr>
<td>AWS-GR_SSM_DOCUMENT_NOT_PUBLIC</td>
<td>il-central-1</td>
</tr>
<tr>
<td>AWS-GR_ROOT_ACCOUNT_MFA_ENABLED</td>
<td>il-central-1, me-central-1</td>
</tr>
<tr>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
<td>il-central-1, eu-south-2, eu-central-2</td>
</tr>
</tbody>
</table>
Regions and stack set limitations

If you plan to extend governance to OUs with a large number of accounts across a large number of AWS Regions, you may encounter limits created by AWS CloudFormation stack sets on the overall size of an organization. You can estimate the limitation with this formula:

\[ \text{Number of managed accounts in Organization} \times \text{Number of governed Regions} \leq 150,000 \]

As a general rule, we expect that the number of accounts supported when extending governance to an OU diminishes with the number of Regions governed.

This limitation becomes apparent if more than 15 Regions where AWS Control Tower is available are activated when you’re extending governance to an OU. The upper limit on the number of accounts per organizational unit (OU) is reduced.

For example, if 22 Regions are activated, the limit is 220 accounts per OU, instead of 300. If you require to extend governance to OUs with more than 220 accounts, you must reduce the number of activated Regions. This reduction is due to stack set limitations.

Guidelines:

- With 15 activated Regions, OUs of up to 300 accounts are supported
- With 22 activated Regions, OUs of up to 220 accounts are supported
- With 16 to 21 activated Regions, the maximum supported OU size is somewhere in the range of 220-300 accounts
- With 23+ activated Regions, the maximum supported OU size is less than 220 accounts

### Control identifier | Unsupported regions
--- | ---
AWS-GR_RDS_STORAGE_ENCRYPTED | eu-south-2
AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK | ap-south-2, eu-south-2
AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK | ap-south-2, eu-south-2
Best practices for AWS Control Tower administrators

This topic is intended primarily for management account administrators.

Management account administrators are responsible for explaining some tasks that AWS Control Tower controls prevent their member account administrators from doing. This topic describes some best practices and procedures for transferring this knowledge, and it gives other tips for setting up and maintaining your AWS Control Tower environment efficiently.

Explaining access to users

The AWS Control Tower console is available only to users with the management account administrator permissions. Only these users can perform administrative work within your landing zone. In accordance with best practices, this means that the majority of your users and member account administrators will never see the AWS Control Tower console. As a member of the management account administrator group, it's your responsibility to explain the following information to the users and administrators of your member accounts, as appropriate.

- Explain which AWS resources that users and administrators have access to within the landing zone.
- List the preventive controls that apply to each organizational unit (OU) so that the other administrators can plan and execute their AWS workloads accordingly.

Explaining resource access

Some administrators and other users may need an explanation of the AWS resources to which they have access to within your landing zone. This access can include programmatic access and console-based access. Generally speaking, read access and write access for AWS resources is allowed. To perform work within AWS, your users require some level of access to the specific services they need to do their jobs.

Some users, such as your AWS developers, may need to know about the resources to which they have access, so they can create engineering solutions. Other users, such as the end users of the applications that run on AWS services, do not need to know about AWS resources within your landing zone.

AWS offers tools to identify the scope of a user’s AWS resource access. After you identify the scope of a user’s access, you can share that information with the user, in accordance with your organization’s information management policies. For more information about these tools, see the links that follow.

- **AWS access advisor** – The AWS Identity and Access Management (IAM) access advisor tool lets you determine the permissions that your developers have by analyzing the last timestamp when an IAM entity, such as a user, role, or group, called an AWS service. You can audit service access and remove unnecessary permissions, and you can automate the process if needed. For more information, see our [AWS Security blog post](https://aws.amazon.com/security/blogs/post/automate-who-can-access-what-with-aws-access-advisor/).
- **IAM policy simulator** – With the IAM policy simulator, you can test and troubleshoot IAM-based and resource-based policies. For more information, see [Testing IAM Policies with the IAM Policy Simulator](https://aws.amazon.com/blogs/security/testing-iam-policies-with-iam-policy-simulator/).
- **AWS CloudTrail logs** – You can review AWS CloudTrail logs to see actions taken by a user, role, or AWS service. For more information about CloudTrail, see the [AWS CloudTrail User Guide](https://docs.aws.amazon.com/AmazonCloudTrail/latest/userguide/).
Actions taken by AWS Control Tower landing zone administrators are viewable in the landing zone management account. Actions taken by member account administrators and users are viewable in the shared log archive account.

You can view a summary table of AWS Control Tower events in the Activities page.

Explaining preventive controls

A preventive control ensures that your organization's accounts maintain compliance with your corporate policies. The status of a preventive control is either enforced or not-enabled. A preventive control prevents policy violations by using service control policies (SCPs). In comparison, a detective control informs you of various events or states that exist, by means of defined AWS Config rules.

Some of your users, such as AWS developers, may need to know about the preventive controls that apply to any accounts and OUs they use, so they can create engineering solutions. The following procedure offers some guidance on how to provide this information for the right users, according to your organization's information management policies.

**Note**
This procedure assumes you've already created at least one child OU within your landing zone, as well as at least one AWS IAM Identity Center user.

**To show preventive controls for users with a need to know**

2. From the left navigation, choose Organization.
3. From the table, choose the name of one of the OUs for which your user needs information about the applicable controls.
4. Note the name of the OU and the controls that apply to this OU.
5. Repeat the previous two steps for each OU about which your user needs information.

For detailed information about the controls and their functions, see About controls in AWS Control Tower (p. 196).

Plan your AWS Control Tower landing zone

When you go through the setup process, AWS Control Tower launches a key resource associated with your account, called a landing zone, which serves as a home for your organizations and their accounts.

**Note**
You can have one landing zone per organization.

For information about some best practices to follow when you plan and set up your landing zone, see AWS multi-account strategy for your AWS Control Tower landing zone (p. 35).

**Ways to Set Up AWS Control Tower**

You can set up an AWS Control Tower landing zone in an existing organization, or you can start by creating a new organization that contains your AWS Control Tower landing zone.

- Launch AWS Control Tower in an Existing Organization (p. 34): This section is for customers who have existing AWS Organizations ready to bring into governance by AWS Control Tower.
Compare functionality

Here's a brief comparison of the differences between adding AWS Control Tower to an existing organization or extending AWS Control Tower governance to OUs and accounts. Also, some special considerations apply if you are moving to AWS Control Tower from the AWS Landing Zone solution.

**About Adding to an Existing Organization:** Adding AWS Control Tower to an existing organization is something you can accomplish within the AWS console. In this case, you've already got an organization that you've created in the AWS Organizations service, that organization is not currently registered with AWS Control Tower, and you want to add a landing zone afterward.

When you add a landing zone to an existing organization, AWS Control Tower sets up a parallel structure, at the AWS Organizations level. It doesn't change the OUs and accounts within your existing organization.

**About Extending Governance:** Extending governance applies to specific OUs and accounts within a single organization that's already registered with AWS Control Tower, which means that a landing zone already exists for that organization. Extending governance means that AWS Control Tower controls are extended so that their constraints apply to the specific OUs and accounts within that registered organization. In this case, you're not launching a new landing zone, you're only expanding the current landing zone for your organization.

**Important**
Special consideration: If you currently are using the AWS Landing Zone solution (ALZ) for AWS Organizations, check with your AWS solutions architect before you try to enable AWS Control Tower in your organization. AWS Control Tower cannot perform pre-checks that determine whether AWS Control Tower may interfere with your current landing zone deployment. For more information, see Walkthrough: Move from ALZ to AWS Control Tower (p. 1429). Also, for information about moving accounts from one landing zone to another, see What if the account does not meet the prerequisites? (p. 112)

Launch AWS Control Tower in an Existing Organization

By setting up an AWS Control Tower landing zone in an existing organization, you can start working immediately, in parallel with your existing AWS Organizations environment. Your other OUs created within AWS Organizations are unchanged, because they are not registered with AWS Control Tower. You can continue to use those OUs and accounts exactly as they are.

AWS Control Tower consolidates by using the management account from your existing organization as its management account. No new management account is needed. You can launch your AWS Control Tower landing zone from your existing management account.

**Note**
To set up AWS Control Tower on an existing organization, your service limits must allow for the creation of at least two additional accounts.

**Effects of adding AWS Control Tower to your existing organization**
AWS Control Tower User Guide

Launch AWS Control Tower in a New Organization

AWS Control Tower creates two accounts in your organization: an audit account and a logging account. These accounts keep a record of actions taken by your team, in their individual end-user accounts. The Audit and Log archive accounts appear in the Security OU within your AWS Control Tower landing zone.

When you set up your landing zone, the accounts added by AWS Control Tower become part of your existing AWS Organizations, and as such they become part of the billing for your existing organization.

Summary of capabilities

Enabling AWS Control Tower on an existing AWS Organizations organization provides several major enhancements to the organization.

- It allows for unified billing across your organization’s groups, because accounts added by AWS Control Tower will become part of your existing organization.
- It gives you the ability to administer all accounts from one management account in your OU.
- It simplifies how you apply and enforce controls that cover security and compliance for existing and new accounts.

Important

Launching your AWS Control Tower landing zone in an existing AWS Organizations organization does not enable you to extend AWS Control Tower governance from that organization to other OUs or accounts that are not registered with AWS Control Tower.

To launch AWS Control Tower in your existing organization, follow the process outlined in Getting started with AWS Control Tower (p. 16).

For more information about how AWS Control Tower interacts with existing AWS Organizations organizations, see Govern organizations and accounts with AWS Control Tower (p. 184).

Launch AWS Control Tower in a New Organization

If you’re new to AWS Control Tower and you haven't worked with AWS Organizations, the best place to begin is with our Setting up (p. 14) document.

AWS Control Tower sets up an organization for you automatically when you don’t have one set up.

AWS multi-account strategy for your AWS Control Tower landing zone

AWS Control Tower customers often seek guidance about how to set up their AWS environment and accounts for best results. AWS has created a unified set of recommendations, called the multi-account strategy, to help you make the best use of your AWS resources, including your AWS Control Tower landing zone.

Essentially, AWS Control Tower acts as an orchestration layer that works with other AWS services, which assist you with implementing the AWS multi-account recommendations for AWS accounts and AWS Organizations. After your landing zone is set up, AWS Control Tower continues to assist you with maintaining your corporate policies and security practices across multiple accounts and workloads.

Most landing zones develop over time. As the number of organizational units (OUs) and accounts in your AWS Control Tower landing zone increases, you can extend your AWS Control Tower deployment in ways that help organize your workloads effectively. This chapter provides prescriptive guidance on how to plan
and set up your AWS Control Tower landing zone, in alignment with the AWS multi-account strategy, and extend it over time.

For a general discussion about best practices for organizational units, see Best Practices for Organizational Units with AWS Organizations.

**AWS multi-account strategy: Best practices guidance**

AWS best practices for a well-architected environment recommend that you should separate your resources and workloads into multiple AWS accounts. You can think of AWS accounts as isolated resource containers: they offer workload categorization, as well as blast radius reduction when things go wrong.

**Definition of an AWS account**

*An AWS account acts as a resource container and resource isolation boundary.*

**Note**

An AWS account is not the same as a user account, which is set up through Federation or AWS Identity and Access Management (IAM).

**More about AWS accounts**

An AWS account provides the ability to isolate resources and to contain security threats for your AWS workloads. An account also provides a mechanism for billing and for governance of a workload environment.

The AWS account is the primary implementation mechanism to provide a resource container for your workloads. If your environment is well-architected, you can manage multiple AWS accounts effectively, and thus, manage multiple workloads and environments.

AWS Control Tower sets up a well-architected environment. It relies upon AWS accounts, along with AWS Organizations, which help govern changes to your environment that can extend across multiple accounts.

**Definition of a well-architected environment**

*AWS defines a well-architected environment as one that begins with a landing zone.*

AWS Control Tower offers a landing zone that is set up automatically. It enforces controls to ensure compliance with your corporate guidelines, across multiple accounts in your environment.

**Definition of a landing zone**

*The landing zone is a cloud environment that offers a recommended starting point, including default accounts, account structure, network and security layouts, and so forth. From a landing zone, you can deploy workloads that utilize your solutions and applications.*

**Guidelines to set up a well-architected environment**

The three key components of a well-architected environment, explained in the following sections, are:

- Multiple AWS accounts
- Multiple organizational units (OUs)
- A well-planned structure
Use multiple AWS accounts

One account isn't enough to set up a well-architected environment. By using multiple accounts, you can best support your security goals and business processes. Here are some benefits of using a multi-account approach:

- **Security controls** – Applications have different security profiles, so they require different control policies and mechanisms. For example, it's far easier to talk to an auditor and point to a single account hosting the payment card industry (PCI) workload.

- **Isolation** – An account is a unit of security protection. Potential risks and security threats can be contained within an account without affecting others. Therefore, security needs may require you to isolate accounts from one another. For example, you may have teams with different security profiles.

- **Many teams** – Teams have different responsibilities and resource needs. By setting up multiple accounts, the teams cannot interfere with one another, as they might when using the same account.

- **Data Isolation** – Isolating data stores to an account helps limit the number of people who have access to data and can manage the data store. This isolation helps prevent unauthorized exposure of highly private data. For example, data isolation helps support compliance with the General Data Protection Regulation (GDPR).

- **Business process** – Business units or products often have completely different purposes and processes. Individual accounts can be established to serve business-specific needs.

- **Billing** – An account is the only way to separate items at a billing level, including things like transfer charges and so forth. The multi-account strategy helps create separate billable items across business units, functional teams, or individual users.

- **Quota allocation** – AWS quotas are set up on a per-account basis. Separating workloads into different accounts gives each account (such as a project) a well-defined, individual quota.

Use multiple organizational units

AWS Control Tower and other account orchestration frameworks can make changes that cross account boundaries. Therefore, the AWS best practices address cross-account changes, which potentially can break an environment or undermine its security. In some cases, changes can affect the overall environment, beyond policies. As a result, we recommend that you should set up at least two mandatory accounts, Production and Staging.

Furthermore, AWS accounts often are grouped into organizational units (OUs), for purposes of governance and control. OUs are designed to handle enforcement of policies across multiple accounts.

Our recommendation is that, at a minimum, you create a pre-production (or Staging) environment that is distinct from your Production environment—with distinct controls and policies. The Production and Staging environments can be created and governed as separate OUs, and billed as separate accounts. In addition, you may want to set up a Sandbox OU for code testing.

Use a well-planned structure for OUs in your landing zone

AWS Control Tower sets up some OUs for you automatically. As your workloads and requirements expand over time, you can extend the original landing zone configuration to suit your needs.

**Note**

The names given in the examples follow the suggested AWS naming conventions for setting up a multi-account AWS environment. You can rename your OUs after you've set up your landing zone, by selecting **Edit** on the OU detail page.

**Recommendations**

After AWS Control Tower sets up the first, required OU for you — the Security OU — we recommend creating some additional OUs in your landing zone.
We recommend that you allow AWS Control Tower to create at least one additional OU, called the Sandbox OU. This OU is for your software development environments. AWS Control Tower can set up the Sandbox OU for you during landing zone creation, if you select it.

Two recommended other OUs you can set up on your own: the Infrastructure OU, to contain your shared services and networking accounts, and an OU to contain your production workloads, called the Workloads OU. You can add additional OUs in your landing zone through the AWS Control Tower console on the Organizational units page.

**Recommended OUs besides the ones set up automatically**

- **Infrastructure OU** – Contains your shared services and networking accounts.
  
  **Note**
  AWS Control Tower does not set up the Infrastructure OU for you.

- **Sandbox OU** – A software development OU. For example, it may have a fixed spending limit, or it may not be connected to the production network.

  **Note**
  AWS Control Tower recommends that you set up the Sandbox OU, but it is optional. It can be set up automatically as part of configuring your landing zone.

- **Workloads OU** – Contains accounts that run your workloads.

  **Note**
  AWS Control Tower does not set up the Workloads OU for you.

For more information see [Production starter organization with AWS Control Tower](#).

**Example of AWS Control Tower with a complete multi-account OU structure**

AWS Control Tower supports a nested OU hierarchy, which means that you can create a hierarchical OU structure that meets your organization's requirements. You can build an AWS Control Tower environment to match the AWS multi-account strategy guidance.

You also can build a simpler, flat OU structure that performs well and aligns with the AWS multi-account guidance. Just because you can build a hierarchical OU structure, it does not mean that you must do so.

- To view a diagram that shows an example set of OUs in an expanded, flat AWS Control Tower environment with AWS multi-account guidance, see [Example: Workloads in a Flat OU Structure](#).

- For more information about how AWS Control Tower works with nested OU structures, see [Nested OUs in AWS Control Tower (p. 186)](#).

- For more information about how AWS Control Tower aligns with the AWS guidance, see the AWS white paper, [Organizing Your AWS Environment Using Multiple Accounts](#).

The diagram on the linked page shows that more Foundational OUs and more Additional OUs have been created. These OUs serve the additional needs of a larger deployment.

In the Foundational OUs column, two OUs have been added to the basic structure:

- **Security_Prod OU** – Provides a read-only area for security policies, as well as a break-glass security audit area.

- **Infrastructure OU** – You may wish to separate the Infrastructure OU, recommended previously, into two OUs, Infrastructure_Test (for pre-production infrastructure) and Infrastructure_Prod (for production infrastructure).
In the Additional OUs area, several more OUs have been added to the basic structure. These following are the next recommended OUs to create as your environment grows:

- **Workloads OU** – The Workloads OU, recommended previously but optional, has been separated into two OUs, Workloads_Test (for pre-production workloads) and Workloads_Prod (for production workloads).
- **PolicyStaging OU** – Allows system administrators to test their changes to controls and policies before fully applying them.
- **Suspended OU** – Offers a location for accounts that may have been disabled temporarily.

**About the Root**

The Root is not an OU. It is a container for the management account, and for all OUs and accounts in your organization. Conceptually, the Root contains all of the OUs. It cannot be deleted. You cannot govern enrolled accounts at the Root level within AWS Control Tower. Instead, govern enrolled accounts within your OUs. For a helpful diagram, see the AWS Organizations documentation.

**Administrative tips for landing zone setup**

- The AWS Region where you do the most work should be your home Region.
- Set up your landing zone and deploy your Account Factory accounts from within your home Region.
- If you’re investing in several AWS Regions, be sure that your cloud resources are in the Region where you’ll do most of your cloud administrative work and run your workloads.
- By keeping your workloads and logs in the same AWS Region, you reduce the cost that would be associated with moving and retrieving log information across regions.
- The audit and other Amazon S3 buckets are created in the same AWS Region from which you launch AWS Control Tower. We recommend that you do not move these buckets.
- You can make your own log buckets in the Log Archive account, but it is not recommended. Be sure to leave the buckets created by AWS Control Tower.
- Your Amazon S3 access logs must be in the same AWS Region as the source buckets.
- When launching, AWS Security Token Service (STS) endpoints must be activated in the management account, for all Regions supported by AWS Control Tower. Otherwise, the launch may fail midway through the configuration process.
- **Do not add tags** to resources in your AWS Control Tower landing zone, because setup will fail. AWS Control Tower does not support tagging.
- We recommend enabling multi-factor authentication (MFA) for every account that AWS Control Tower manages.

**Considerations about VPCs**

- The VPC created by AWS Control Tower is limited to the AWS Regions in which AWS Control Tower is available. Some customers whose workloads run in non-supported Regions may want to disable the VPC that is created with your Account Factory account. They may prefer to create a new VPC using the Service Catalog portfolio, or to create a custom VPC that runs only in the required Regions.
- The VPC created by AWS Control Tower is not the same as the default VPC that is created for all AWS accounts. In Regions where AWS Control Tower is supported, AWS Control Tower deletes the default VPC when it creates the AWS Control Tower VPC.
- If you delete your default VPC in your home AWS Region, it’s best to delete it in all other AWS Regions.
Recommendations for setting up groups, roles, and policies

As you set up your landing zone, it's a good idea to decide ahead of time which users will require access to certain accounts and why. For example, a security account should be accessible only to the security team, the management account should be accessible only to the cloud administrators' team, and so forth.

For more information about this topic, see Identity and access management in AWS Control Tower (p. 1387).

Recommended restrictions

You can restrict the scope of administrative access to your organizations by setting up an IAM role or policy that allows administrators to manage AWS Control Tower actions only. The recommended approach is to use the IAM policy arn:aws:iam::aws:policy/service-role/AWSControlTowerServiceRolePolicy. With the AWSControlTowerServiceRolePolicy role enabled, an administrator can manage AWS Control Tower only. Be sure to include appropriate access to AWS Organizations for managing your preventive controls, and SCPs, and access to AWS Config, for managing detective controls, in each account.

When you're setting up the shared audit account in your landing zone, we recommend that you assign the AWSSecurityAuditors group to any third-party auditors of your accounts. This group gives its members read-only permission. An account must not have write permissions on the environment that it is auditing, because it can violate compliance with Separation of Duty requirements for auditors.

You can impose conditions in your role trust policies, to restrict the accounts and resources that interact with certain roles in AWS Control Tower. We strongly recommend that you restrict access to the AWSControlTowerAdmin role, because it allows wide access permissions. For more information, see Optional conditions for your role trust relationships (p. 88).

Guidance for creating and modifying AWS Control Tower resources

We recommend the following best practices as you create and modify resources in AWS Control Tower. This guidance might change as the service is updated.

General Guidance

- Do not modify or delete resources created by AWS Control Tower in the management account or in the shared accounts. Modification of these resources can require you to update your landing zone or re-register an OU.

- Do not modify or delete the AWS Identity and Access Management (IAM) roles created within the shared accounts in the Security organizational unit (OU). Modification of these roles can require an update to your landing zone.

- For more information about the resources created by AWS Control Tower, see What Are the Shared Accounts? (p. 3).

- Do not disallow usage of any AWS Regions through either SCPs or AWS Security Token Service (AWS STS). Doing so will cause AWS Control Tower to enter an undefined state. If you disallow Regions with AWS STS, your functionality will fail in those Regions, because authentication would be unavailable in
When to sign in as a root user

Certain administrative tasks require that you must sign in as a root user. You can sign in as a root user to an AWS account that was created by account factory in AWS Control Tower.
You must sign in as a root user to perform the following actions:

- Change certain account settings, including the account name, root user password, or email address. For more information, see Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog (p. 123).
- To close an AWS account.
- For more information about actions that require root user login credentials, see Tasks that require root user credentials in the AWS Account Management Reference Guide.

Note
To change or enable your AWS Support plan, you must be signed in as the root user or be a user with the appropriate IAM permissions.

To sign in as root user
1. Open the AWS sign-in page.
   If you don't have the email address of the AWS account to which you require access, you can get it from AWS Control Tower. Open the console for the management account, choose Accounts, and look for the email address.
2. Enter the email address of the AWS account to which you require access, and then choose Next.
3. Choose Forgot password? to have password reset instructions sent to the root user email address.
4. Open the password reset email message from the root user mailbox, then follow the instructions to reset your password.
5. Open the AWS sign-in page, then sign in with your reset password.

AWS Organizations Guidance

- You can find guidance about best practices to protect the security of your AWS Control Tower management account and member accounts in the AWS Organizations documentation.
  - Best practices for the management account
  - Best practices for member accounts
- Don't use AWS Organizations to update service control policies (SCPs) attached to an OU that is registered with AWS Control Tower. Doing so could result in the controls entering an unknown state, which will require you to repair your landing zone or re-register your OU in AWS Control Tower. Instead, you can create new SCPs and attach those to the OUs rather than editing the SCPs that AWS Control Tower has created.
- Moving individual, already enrolled, accounts into AWS Control Tower, from outside of a registered OU, causes drift that must be repaired. See Types of Governance Drift (p. 172).
- If you use AWS Organizations to create, invite, or move accounts within an organization registered with AWS Control Tower, those accounts are not enrolled by AWS Control Tower and those changes are not recorded. If you need access to these accounts through SSO, see Member Account Access.
- If you use AWS Organizations to move an OU into an organization created by AWS Control Tower, the external OU is not registered by AWS Control Tower.
- AWS Control Tower handles permission filtering differently than AWS Organizations does. If your accounts are provisioned with AWS Control Tower account factory, end-users can see the names and parents of all OUs in the AWS Control Tower console, even if they don't have permission to retrieve those names and parents from AWS Organizations directly.
- AWS Control Tower does not support mixed permissions on organizations, such as permission to view an OU's parent but not to view OU names. For this reason, AWS Control Tower administrators are expected to have full permissions.
• The AWS Organizations FullAWSAccess SCP must be applied and should not be merged with other SCPs. Change to this SCP is not reported as drift; however, some changes may affect AWS Control Tower functionality in unpredictable ways, if access to certain resources is denied. For example, if the SCP is detached, or modified, an account may lose access to an AWS Config recorder or create a gap in CloudTrail logging.

• Don’t use the AWS Organizations DisableAWSServiceAccess API to turn off AWS Control Tower service access to the organization where you’ve set up your landing zone. If you do so, certain AWS Control Tower drift detection features may not function properly without messaging support from AWS Organizations. These drift detection features help guarantee that AWS Control Tower can report the compliance status of organizational units, accounts, and controls in your organization accurately. For more information, see API_DisableAWSServiceAccess in the AWS Organizations API Reference.

IAM Identity Center guidance

Note
SSO is an abbreviation used in the technology industry to denote single sign-on. In general terms, SSO is a session and user authentication service. It permits someone to use one set of login credentials for access to many applications. When referring to the single-sign on capability in AWS, we are referring to the AWS service called AWS Identity and Access Management, and abbreviated as IAM or IAM Identity Center.

AWS Control Tower recommends that you use AWS Identity and Access Management (IAM) to regulate access to your AWS accounts. However, you have the option to choose whether AWS Control Tower sets up IAM Identity Center for you, whether you set up IAM Identity Center for yourself, in a way that meets your business requirements most effectively, or whether to select another method for account access.

By default, AWS Control Tower sets up AWS IAM Identity Center for your landing zone, in alignment with best-practices guidance defined in Organizing your AWS environment using multiple accounts. Most customers choose the default. Alternative access methods are required sometimes, for regulatory compliance in specific industries or countries, or in AWS Regions where AWS IAM Identity Center is not available.

Choosing an option

From the console, you can choose to self-manage IAM Identity Center during the landing zone set up process, rather than allowing AWS Control Tower to set it up for you. At any time later, you can choose to change this selection, by modifying the landing zone settings and updating your landing zone on the landing zone Settings page.

To discontinue AWS IAM Identity Center in AWS Control Tower, or to begin using AWS IAM Identity Center

1. Navigate to the landing zone Settings page
2. Select the Configurations tab
3. Then choose the appropriate radio button, to change your selection for AWS IAM Identity Center.

After you choose to self-manage AWS IAM Identity Center as your IdP, AWS Control Tower creates only those roles and policies needed to manage AWS Control Tower, such as AWSControlTowerAdmin and AWSControlTowerAdminPolicy. For landing zones that self-manage, AWS Control Tower no longer creates IAM roles and groupings for customer-specific use — not during the landing zone set-up process, nor during account provisioning with Account Factory.

Note
If you remove AWS IAM Identity Center from your AWS Control Tower landing zone, the users, groups, and permission sets that AWS Control Tower created are not removed. We recommend that you remove these resources.
Account Factory customers with alternative identity providers (IdPs) such as Azure AD, Ping, or Okta, can follow the AWS IAM Identity Center process to connect to an external identity provider and onboard their IdP. You can return to having AWS Control Tower generate your groupings and roles at any time, by modifying the landing zone settings.

- For specific information about how AWS Control Tower works with IAM Identity Center based on your identity source, see Considerations for AWS IAM Identity Center customers in the Pre-launch checks section of the Getting Started page of this User Guide.
- For additional information about how the behavior of AWS Control Tower interacts with IAM Identity Center and different identity sources, refer to Considerations for Changing Your Identity Source in the IAM Identity Center User Guide.
- See Manage Users and Access Through AWS IAM Identity Center (p. 1380) for more information about working with AWS Control Tower and IAM Identity Center.

**Account Factory guidance**

You can encounter issues when using Account Factory to provision a new account in AWS Control Tower. For information about how to troubleshoot these issues, see the section New Account Provisioning Failed (p. 1446) in Troubleshooting of the AWS Control Tower User Guide.

We recommend that you create federated users or IAM roles instead of IAM users. Federated users and IAM roles provide you with temporary credentials. IAM users have long-term credentials that can be difficult to manage. For more information, see IAM identities (users, user groups, and roles) in the IAM User Guide.

If you're authenticated as an IAM user or IAM Identity Center user when provisioning a new account in Account Factory or when using the Enroll account feature AWS Control Tower, verify that your user has access to your AWS Service Catalog portfolio. Otherwise, you might receive an error message from Service Catalog. For more information, see No Launch Paths Found Error (p. 1449) in the Troubleshooting section of the AWS Control Tower User Guide.

**Note**
Up to five accounts can be provisioned at a time.

**Guidance on Subscribing to SNS Topics**

- The aws-controltower-AllConfigNotifications SNS topic receives all events published by AWS Config, including compliance notifications and Amazon CloudWatch event notifications. For example, this topic informs you if a control violation has occurred. It also gives information about other types of events. (Learn more from AWS Config about what they publish when this topic is configured.)
- Data Events from the aws-controltower-BaselineCloudTrail trail are set to publish to the aws-controltower-AllConfigNotifications SNS topic as well.
- To receive detailed compliance notifications, we recommend that you subscribe to the aws-controltower-AllConfigNotifications SNS topic. This topic aggregates compliance notifications from all child accounts.
- To receive drift notifications and other notifications as well as compliance notifications, but fewer notifications overall, we recommend that you subscribe to the aws-controltower-AggregateSecurityNotifications SNS topic.
- To receive notifications about AWS Control Tower Account Factory for Terraform (AFT) errors, you can subscribe to an SNS topic called aft_failure_notifications, shown in the AFT repository. For example:

```hcl
resource "aws_sns_topic" "aft_failure_notifications" {
```
name = "aft-failure-notifications"
kms_master_key_id = "alias/aws/sns"

Guidance for KMS keys

AWS Control Tower works with AWS Key Management Service (AWS KMS). Optionally, if you wish to encrypt and decrypt your AWS Control Tower resources with an encryption key that you manage, you can generate and configure AWS KMS keys. You can add or change a KMS key any time you update your landing zone. As a best practice, we recommend using your own KMS keys and changing them from time to time.

AWS KMS allows you to create multi-Region KMS keys and asymmetric keys. However, AWS Control Tower does not support multi-Region keys or asymmetric keys. AWS Control Tower performs a pre-check of your existing keys. You may see an error message if you select a multi-Region key or an asymmetric key. In that case, generate another key for use with AWS Control Tower resources.

For customers who operate an AWS CloudHSM cluster: Create a custom key store associated with your CloudHSM cluster. Then you can create a KMS key, which resides in the CloudHSM custom key store you created. You can add this KMS key to AWS Control Tower.

You must make a specific update to the permissions policy of a KMS key to make it work with AWS Control Tower. For details, refer to the section called Update the KMS key policy (p. 23).
Configuration update management in AWS Control Tower

It is the responsibility of the members of your central cloud administrators' team to keep your landing zone updated. Updating your landing zone ensures that AWS Control Tower is patched and updated. In addition, to protect your landing zone from potential compliance issues, the members of the central cloud administrator team should resolve drift issues as soon as they're detected and reported.

**Note**
The AWS Control Tower console indicates when your landing zone needs to be updated. If you don't see an option to update, your landing zone is already up to date.

The following table contains a list of AWS Control Tower landing zone update releases, with links to descriptions of each release.

<table>
<thead>
<tr>
<th>Version</th>
<th>Release date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>6-09-2023</td>
<td>Landing zone version 3.2 (p. 1462)</td>
</tr>
<tr>
<td>3.1</td>
<td>2-09-2023</td>
<td>Landing zone version 3.1 (p. 1468)</td>
</tr>
<tr>
<td>3.0</td>
<td>7-26-2022</td>
<td>Landing zone version 3.0 (p. 1473)</td>
</tr>
<tr>
<td>2.9</td>
<td>4-22-2022</td>
<td>Landing zone version 2.9 (p. 1477)</td>
</tr>
<tr>
<td>2.8</td>
<td>2-10-2022</td>
<td>Landing zone version 2.8 (p. 1477)</td>
</tr>
<tr>
<td>2.7</td>
<td>4-8-2021</td>
<td>Landing zone version 2.7</td>
</tr>
<tr>
<td>2.6</td>
<td>12-29-2020</td>
<td>Landing zone version 2.6</td>
</tr>
<tr>
<td>2.5</td>
<td>11-18-2020</td>
<td>Landing zone version 2.5</td>
</tr>
<tr>
<td>2.4</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>2.3</td>
<td>3-5-2020</td>
<td>Landing zone version 2.3</td>
</tr>
<tr>
<td>2.2</td>
<td>11-13-19</td>
<td>Landing zone version 2.2</td>
</tr>
<tr>
<td>2.1</td>
<td>6-24-19</td>
<td>Landing zone version 2.1</td>
</tr>
</tbody>
</table>

Each time you update your landing zone, you have the opportunity to modify your landing zone settings.

**Benefits of updating**

- You can change your governed Regions
• You can change your log retention policy
• You can add or remove the Region deny control
• You can apply AWS KMS encryption keys
• You can activate or deactivate your organization-level CloudTrail trail.
• You can resolve landing zone drift (p. 172)

When you update your landing zone, you receive the latest features for AWS Control Tower, automatically. View your current landing zone version on the Landing zone settings page.

You have the opportunity to clear unused AWS Identity center (formerly called AWS SSO) mappings when you update your landing zone. For more information, see Field Notes: Clear Unused IAM Identity Center Mappings Automatically During AWS Control Tower Upgrades.

Prerequisite for Update and Repair – turn off Requester Pays
Before you update or repair your landing zone, be sure that the Amazon S3 logging bucket for the Log Archive account does not have the Requester Pays feature enabled. You must turn off that feature before you begin the Update or Repair process. When AWS Control Tower sets up your logging bucket, this feature is not enabled. Therefore, only the customers who have subsequently activated the Requester Pays feature must turn it off. For more information, see Amazon S3 bucket policy for CloudTrail and Using Requester Pays buckets.

About Updates

Updates are required to correct governance drift, or to move to a new version of AWS Control Tower. To perform a complete update of AWS Control Tower, you must update your landing zone first and then update the enrolled accounts individually. You may need to perform three types of updates at different times.

• A landing zone update: Most often this type of update is performed by choosing Update on the Landing zone settings page. You may need to perform a landing zone update to repair certain types of drift, and you can choose Repair when necessary.

• An update of one or more individual accounts: You must update accounts if the associated information changes, or if certain types of drift have occurred. If an account requires an update, the account's status will show Update available on the Accounts page.

To update a single account, navigate to the account detail page and select Update account. Accounts also may be updated by a manual process, by choosing Re-register OU, or with an automated scripting approach, described in a later section of this page.

• A full update: A full update includes an update of your landing zone, followed by an update of all the enrolled accounts in your registered OU. Full updates are required with a new release of AWS Control Tower such as 2.9, 3.0, and so forth.

Note
After completing a landing zone update, you cannot undo the update or downgrade to a previous version.

Update Your Landing Zone

The easiest way to update your AWS Control Tower landing zone is through the Landing zone settings page, which you can reach by choosing Landing zone settings in the left navigation of the AWS Control Tower dashboard.
The **Landing zone settings** page shows you the current version of your landing zone, and it lists any updated versions that may be available. You can choose the **Update** button if you need to update your version.

**Note**
Alternatively, you can update your landing zone manually. The update takes approximately the same amount of time, whether you use the **Update** button or the manual process. To perform a manual update of your landing zone only, see steps 1 and 2 that follow.

## Manual updates

The following procedure walks you through the steps of a full update for AWS Control Tower manually. To update an individual account, see [Update the account in the console](p. 123).

### To update your landing zone manually, with any number of accounts per OU

2. Review the information in the wizard and choose **Update**. This updates the backend of the landing zone as well as your shared accounts. This process can take a little more than half an hour.
3. Update your member accounts (this procedure must be followed for an OU that contains over 300 accounts).
4. From the left navigation pane, choose **Organization**.
5. To update each account, follow the steps given in [Update the account in the console](p. 123).

### Optionally Re-register OU to update accounts

For registered AWS Control Tower OUs with fewer than 300 accounts, you can go to the OU page in the dashboard and select **Re-register OU** to update the accounts in that OU.

## Resolve drift with Repair and Re-register

Drift often occurs as you and your organization members use the landing zone.

Drift detection is automatic in AWS Control Tower. Automated scans of your SCPs help you identify resources that need changes or configuration updates that must be made to resolve the drift.

To repair most types of drift, choose **Repair** on the **Landing zone settings** page. Also, you can repair some types of drift by choosing to **Re-register** an OU. For more information about types of drift and how to resolve them, see [Types of Governance Drift](p. 172) and [Detect and resolve drift in AWS Control Tower](p. 169).

One special case of repair occurs for **role drift**. If a required role is not available, the console shows a warning page and some instructions on how to restore the role. Your landing zone is unavailable until the role drift is repaired. This drift repair is not the same as a full landing zone repair. For more information, see [Don't delete required roles](p. 171).

**Note**
When you fully repair your landing zone, the landing zone is upgraded to the latest landing zone version.

## Provision and update accounts using automation

You can provision or update individual accounts in AWS Control Tower by several methods:
• You can provision and customize accounts with **AWS Control Tower Account Factory for Terraform (AFT)**. For more information, see [Overview of AWS Control Tower Account Factory for Terraform (AFT)](p. 144).

• You can update accounts with **Customizations for AWS Control Tower (CfCT)**. For more information, see [Customizations for AWS Control Tower (CfCT) overview](p. 58).

• **Script automation**: If you prefer to use an API approach, you can update accounts using the API framework of Service Catalog and the AWS CLI to update the accounts in a batch process. You’d call the `UpdateProvisionedProduct` API of Service Catalog for each account. You can write a script to update the accounts, one by one, with this API. More information about this approach, when adding Regions for governance, is available in a blog post, [Enabling guardrails in new AWS Regions](p. 1429).

  You can update as many as five (5) accounts at a time. You must wait for at least one account update to succeed before beginning the next account update. Therefore, the process may take a long time if you have a lot of accounts, but it is not complicated. For more information about this approach, see the [Walkthrough: Automate Account Provisioning in AWS Control Tower by Service Catalog APIs](p. 1429).

**Video walkthrough**

The [Video Walkthrough](p. 1432) is designed for automated account provisioning with a script, but the steps also apply to account updating. Use the `UpdateProvisionedProduct` API instead of the `ProvisionProduct` API.

A further step of automation by script is to check for **Succeed** status of the AWS Control Tower `UpdateLandingZone` lifecycle event. Use it as a trigger to begin updating individual accounts as described in the video. A lifecycle event marks the completion of a sequence of activities, so the occurrence of this event means that a landing zone update is complete. The landing zone update must be complete before account updates begin. For more information about working with lifecycle events, see [Lifecycle Events](p. 147).

**Also see:**

• [Using AWS CloudShell to work with AWS Control Tower](p. 51).

• [Automate tasks in AWS Control Tower](p. 50).
Automate tasks in AWS Control Tower

Many customers prefer to automate tasks in AWS Control Tower, such as account provisioning, control assignment, and auditing. You can set up these automated actions with calls to:

- AWS Service Catalog APIs
- AWS Organizations APIs
- AWS Control Tower APIs
- the AWS CLI

The Related information (p. 1453) page contains links to many excellent technical blog posts that can help you automate tasks in AWS Control Tower. The sections that follow provide links to areas in this AWS Control Tower User Guide that can assist you with automating tasks.

Automating control tasks

You can automate tasks related to applying and removing controls (also known as guardrails) through the AWS Control Tower API. For details, see the AWS Control Tower API Reference.

For more information about how to perform operations with AWS Control Tower APIs, see the blog post AWS Control Tower releases API, pre-defined controls to your organizational units.

Automated account closure

You can automate the closure of AWS Control Tower member accounts with an AWS Organizations API. For more information, see Close an AWS Control Tower member account through AWS Organizations (p. 127).

Automated account provisioning and updating

AWS Control Tower Account Factory Customization (AFC) helps you create accounts from the AWS Control Tower console, with customized AWS CloudFormation templates that we refer to as blueprints. This process is automated in the sense that you can create new accounts and update accounts repeatedly, after setting up a single blueprint, without maintaining pipelines.

AWS Control Tower Account Factory for Terraform (AFT) follows a GitOps model to automate the processes of account provisioning and account updating in AWS Control Tower. For more information, see Provision accounts with AWS Control Tower Account Factory for Terraform (AFT) (p. 139).

Customizations for AWS Control Tower (CfCT) helps you customize your AWS Control Tower landing zone and stay aligned with AWS best practices. Customizations are implemented with AWS CloudFormation templates and service control policies (SCPs). For more information, see Customizations for AWS Control Tower (CfCT) overview (p. 58).

For more information and a video about automated account provisioning, see Walkthrough: Automated account provisioning in AWS Control Tower and Automated provisioning with IAM roles.

Also see Update accounts by script.

Programmatic auditing of accounts
For more information about auditing accounts programmatically, see Programmatic roles and trust relationships for the AWS Control Tower audit account.

**Automating other tasks**

For information about how to increase certain AWS Control Tower service quotas with an automated request method, view this video: Automate Service Limit Increases.

For technical blogs that cover automation and integration use cases, see Automation and integration.

Two open source samples are available on GitHub to help you with certain automation tasks related to security.

- The sample called aws-control-tower-org-setup-sample shows how to automate setting up the Audit account as the delegated administrator for security-related services.
- The sample called aws-control-tower-account-setup-using-step-functions shows how to automate security best practices using Step Functions, when provisioning and configuring new accounts. This sample includes adding principals to organizationally-shared AWS Service Catalog portfolios and associating organization-wide AWS IAM Identity Center groups to new accounts automatically. It also illustrates how to delete the default VPC in every Region.

The AWS Security Reference Architecture includes code examples for automating tasks related to AWS Control Tower. For more information, see the AWS Prescriptive Guidance pages and the associated GitHub repository.

For information about using AWS Control Tower with AWS CloudShell, an AWS service that facilitates working in the AWS CLI, see AWS CloudShell and the AWS CLI.

Because AWS Control Tower is an orchestration layer for AWS Organizations, many other AWS services are available by means of APIs and the AWS CLI. For more information, see Related AWS services.

---

**Using AWS CloudShell to work with AWS Control Tower**

AWS CloudShell is an AWS service that facilitates working in the AWS CLI — it's a browser-based, pre-authenticated shell that you can launch directly from the AWS Management Console. There's no need to download or install command line tools. You can run AWS CLI commands for AWS Control Tower and other AWS services from your preferred shell (Bash, PowerShell or Z shell).

When you launch AWS CloudShell from the AWS Management Console, the AWS credentials you used to sign in to the console are available in a new shell session. You can skip entering your configuring credentials when you interact with AWS Control Tower and other AWS services, and you'll be using AWS CLI version 2, which is pre-installed on the shell's compute environment. You're pre-authenticated with AWS CloudShell.

**Obtaining IAM permissions for AWS CloudShell**

AWS Identity and Access Management provides access management resources that allow administrators to grant permissions to IAM users and IAM Identity Center users for access to AWS CloudShell.

The quickest way for an administrator to grant access to users is through an AWS managed policy. An AWS managed policy is a standalone policy that's created and administered by AWS. The following AWS managed policy for CloudShell can be attached to IAM identities:
• **AWSCloudShellFullAccess**: Grants permission to use AWS CloudShell with full access to all features.

If you want to limit the scope of actions that an IAM user or IAM Identity Center user can perform with AWS CloudShell, you can create a custom policy that uses the AWSCloudShellFullAccess managed policy as a template. For more information about limiting the actions that are available to users in CloudShell, see [Managing AWS CloudShell access and usage with IAM policies](#) in the *AWS CloudShell User Guide*.

**Note**
Your IAM identity also requires a policy that grants permission to make calls to AWS Control Tower. For more information, see [Permissions required to use the AWS Control Tower console](#).

### Interacting with AWS Control Tower using AWS CloudShell

After you launch AWS CloudShell from the AWS Management Console, you can immediately start to interact with AWS Control Tower from the command line interface. AWS CLI commands work in the standard way in CloudShell.

**Note**
When using AWS CLI in AWS CloudShell, you don't need to download or install any additional resources. You're already authenticated within the shell, so you don't need to configure credentials before making calls.

**Launch AWS CloudShell**

- From the AWS Management Console, you can launch CloudShell by choosing the following options available on the navigation bar:
  
  - Choose the CloudShell icon.
  - Start typing "cloudshell" in Search box and then choose the CloudShell option.

Now that you've started CloudShell, you can enter any AWS CLI commands you require to work with AWS Control Tower. For example, you can check your AWS Config status.

**Using AWS CloudShell to help set up AWS Control Tower**

Before performing these procedures, unless it's otherwise indicated, you must be signed in to the AWS Management Console in the home Region for your landing zone, and you must be signed in as an IAM Identity Center user or IAM user with administrative permissions for the management account that contains your landing zone.

1. Here's how you can use AWS Config CLI commands in AWS CloudShell to determine the status of your configuration recorder and delivery channel before you start to configure your AWS Control Tower landing zone.

**Check your AWS Config status**

**View commands:**

- `aws configservice describe-delivery-channels`
- `aws configservice describe-delivery-channel-status`
- `aws configservice describe-configuration-recorders`
- The normal response is something like "name": "default"
2. If you have an existing AWS Config recorder or delivery channel that you need to delete before you set up your AWS Control Tower landing zone, here are some commands you can enter:

**Manage your pre-existing AWS Config resources**

**Delete commands:**

- `aws configservice stop-configuration-recorder --configuration-recorder-name NAME-FROM-DESCRIBE-OUTPUT`
- `aws configservice delete-delivery-channel --delivery-channel-name NAME-FROM-DESCRIBE-OUTPUT`
- `aws configservice delete-configuration-recorder --configuration-recorder-name NAME-FROM-DESCRIBE-OUTPUT`

**Important**

Do not delete the AWS Control Tower resources for AWS Config. Loss of these resources can cause AWS Control Tower to enter an inconsistent state.

**For more information, see the AWS Config documentation**

- [Managing the Configuration Recorder (AWS CLI)](#)
- [Managing the Delivery Channel](#)

3. This example shows AWS CLI commands you'd enter from AWS CloudShell to enable or disable trusted access for AWS Organizations. For AWS Control Tower you do not need to enable or disable trusted access for AWS Organizations, it is just an example. However, you may need to enable or disable trusted access for other AWS services if you're automating or customizing actions in AWS Control Tower.

**Enable or disable trusted service access**

- `aws organizations enable-aws-service-access`
- `aws organizations disable-aws-service-access`

**Create an Amazon S3 bucket with AWS CloudShell**

In the following example, you can use AWS CloudShell to create an Amazon S3 bucket and then use the `PutObject` method to add a code file as an object in that bucket.

1. To create a bucket in a specified AWS Region, enter the following command in the CloudShell command line:

   ```
   aws s3api create-bucket --bucket insert-unique-bucket-name-here --region us-east-1
   ```

   If the call is successful, the command line displays a response from the service similar to the following output:

   ```
   {
   "Location": "/insert-unique-bucket-name-here"
   }
   ```

   **Note**

   If you don't adhere to the rules for naming buckets (using only lowercase letters, for example), the following error is displayed: An error occurred (InvalidBucketName) when calling the CreateBucket operation: The specified bucket is not valid.
To upload a file and add it as an object to the bucket that was just created, call the **PutObject** method:

```
aws s3api put-object --bucket insert-unique-bucket-name-here --key add_prog --body add_prog.py
```

If the object is uploaded successfully to the Amazon S3 bucket, the command line displays a response from the service similar to the following output:

```
{
  "ETag": "\"ab123c1:1:w:wad4a567d8bdf9a1234ebbea56\""
}
```

The ETag is the hash of the object that's been stored. It can be used to check the integrity of the object uploaded to Amazon S3.

### Creating AWS Control Tower resources with AWS CloudFormation

AWS Control Tower is integrated with AWS CloudFormation, a service that helps you to model and set up your AWS resources so that you can spend less time creating and managing your resources and infrastructure. You create a template that describes all the AWS resources that you want, such as `AWS::ControlTower::EnabledControl` (such as controls). AWS CloudFormation provisions and configures those resources for you.

When you use AWS CloudFormation, you can reuse your template to set up your AWS Control Tower resources consistently and repeatedly. Describe your resources once, and then provision the same resources over and over in multiple AWS accounts and Regions.

### AWS Control Tower and AWS CloudFormation templates

To provision and configure resources for AWS Control Tower and related services, you must understand [AWS CloudFormation templates](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/what-is-cloudformation-templates.html). Templates are formatted text files in JSON or YAML. These templates describe the resources that you want to provision in your AWS CloudFormation stacks. If you're unfamiliar with JSON or YAML, you can use AWS CloudFormation Designer to help you get started with AWS CloudFormation templates. For more information, see [What is AWS CloudFormation Designer?](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/what-is-cloudformation-designer.html) in the [AWS CloudFormation User Guide](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/).  

AWS Control Tower supports creating `AWS::ControlTower::EnabledControl` (control resources) in AWS CloudFormation. For more information, including examples of JSON and YAML templates for `AWS::ControlTower::EnabledControl`, see the [AWS Control Tower resource type reference](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-controltower-enabledcontrol.html) in the [AWS CloudFormation User Guide](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/).

**Note**  
The limit for `EnableControl` and `DisableControl` updates in AWS Control Tower is 10 concurrent operations.

To view some AWS Control Tower examples for the CLI and the console, see [Enable controls with AWS CloudFormation](p. 207).

### Learn more about AWS CloudFormation

To learn more about AWS CloudFormation, see the following resources:
• AWS CloudFormation
• AWS CloudFormation User Guide
• AWS CloudFormation API Reference
• AWS CloudFormation Command Line Interface User Guide
Customize your AWS Control Tower landing zone

Certain aspects of your AWS Control Tower landing zone are configurable in the console, such as selection of Regions and optional controls. Other changes may be made outside the console, with automation.

For example, you can create more extensive customizations of your landing zone with the Customizations for AWS Control Tower capability, a GitOps-style customization framework that works with AWS CloudFormation templates and AWS Control Tower lifecycle events.

Customize from the AWS Control Tower console

To make these customizations to your landing zone, follow the steps given by the AWS Control Tower console.

Select customized names during setup

- You can select your top-level OU names during setup. You can rename your OUs at any time using the AWS Organizations console, but making changes to your OUs in AWS Organizations may cause repairable drift (p. 169).
- You can select the names of your shared Audit and Log Archive accounts, but you cannot change the names after setup. (This is a one-time selection.)

Tip
Remember that renaming an OU in AWS Organizations does not update the corresponding provisioned product in Account Factory. To update the provisioned product automatically (and avoid drift), you must perform the OU operation through AWS Control Tower, including creating, deleting, or re-registering an OU.

Select AWS Regions

- You can customize your landing zone by selecting specific AWS Regions for governance. Follow the steps in the AWS Control Tower console.
- You can select and de-select AWS Regions for governance when you update your landing zone.
- You can set the Region Deny control to Enabled or Not enabled, and control user access to most AWS services in ungoverned AWS Regions.

For information about AWS Regions where CfCT has deployment limitations, see Control limitations (p. 28).

Customize by adding optional controls

- Strongly recommended and elective controls are optional, which means that you can customize the level of enforcement for your landing zone by choosing which ones to enable. Optional controls (p. 1360) are not enabled by default.
- The optional Controls that enhance data residency protection (p. 1343) allow you to customize the Regions in which you store and allow access to your data.
• The optional controls that are part of the integrated Security Hub standard allow you to scan your
AWS Control Tower environment to check for security risks.

• The optional proactive controls allow you to check your AWS CloudFormation resources before
they are provisioned, to make sure the new resources will comply with your environment's control
objectives.

Customize your AWS CloudTrail trails

• When you update your landing zone to version 3.0 or later, you can choose to opt into or opt out of
organization-level CloudTrail trails managed by AWS Control Tower. You can change this selection
any time you update your landing zone. AWS Control Tower creates an organization-level trail in your
management account, and that trail enters active or inactive status, based on your choice. Landing
zone 3.0 does not support account-level CloudTrail trails; however, if you require these, you can
configure and manage your own trails. You may incur additional cost for duplicate trails.

Create customized member accounts in the console

• You can create AWS Control Tower member accounts that are customized, and you can update existing
member accounts to add customizations, from the AWS Control Tower console. For more information,
see Customize accounts with Account Factory Customization (AFC) (p. 129).

Automate customizations outside the AWS Control Tower console

Some customizations are not available through the AWS Control Tower console, but they can be
implemented in other ways. For example:

• You can customize accounts during provisioning, in a GitOps-style workflow, with Account Factory for
Terraform (AFT) (p. 139).

AFT is deployed with a Terraform module, available in the AFT repository.

• You can customize your AWS Control Tower landing zone with Customizations for AWS Control
Tower (p. 58) (CfCT), a package of functionality that is built upon AWS CloudFormation templates
and service control policies (SCPs). You can deploy the custom templates and policies to individual
accounts and organizational units (OUs) within your organization.

Source code for CfCT is available in a GitHub repository.

Benefits of Customizations for AWS Control Tower (CfCT)

The package of functionality that we refer to as Customizations for AWS Control Tower (CfCT) helps you
create more extensive customizations for your landing zone than you can create in the AWS Control
Tower console. It offers a GitOps-style, automated process. You can reshape your landing zone to meet
your business requirements.

This infrastructure-as-code customization process integrates AWS CloudFormation templates with AWS
service control policies (SCPs) and AWS Control Tower lifecycle events (p. 1416), so that your resource
deployments remain synchronized with your landing zone. For example, when you create a new account
with Account Factory, the resources attached to the account and the OU can be deployed automatically.
Note
Unlike Account Factory and AFT, CfCT is not specifically intended to create new accounts, but to customize accounts and OUs in your landing zone by deploying resources that you specify.

Benefits

• **Expand a customized and secure AWS environment** – You can expand your multi-account AWS Control Tower environment more quickly, and incorporate AWS best practices into a repeatable customization workflow.

• **Instantiate your requirements** – You can customize your AWS Control Tower landing zone for your business requirements, with the AWS CloudFormation templates and service control policies that express your policy intentions.

• **Automate further with AWS Control Tower lifecycle events** – Lifecycle events allow you to deploy resources based on completion of a previous series of events. You can rely on a lifecycle event to help you deploy resources to accounts and OUs, automatically.

• **Extend your network architecture** – You can deploy customized network architectures that improve and protect your connectivity, such as a transit gateway.

Additional CfCT examples

• An example networking use case with *Customizations for AWS Control Tower* (CfCT) is given in the AWS Architecture blog post, [Deploy consistent DNS with Service Catalog and AWS Control Tower customizations](https://aws.amazon.com/blogs/architecture/deploy-consistent-dns-with-service-catalog-and-aws-control-tower-customizations/).

• A specific example related to CfCT and Amazon GuardDuty is available on GitHub in the [aws-samples repository](https://github.com/aws-samples/). Many of these examples contain sample manifest.yaml files in a directory named customizations_for_aws_control_tower.

For more information about the AWS Security Reference Architecture, see the [AWS Prescriptive Guidance pages](https://aws.amazon.com/prescriptive-guidance/).

Customizations for AWS Control Tower (CfCT) overview

*Customizations for AWS Control Tower* (CfCT) helps you customize your AWS Control Tower landing zone and stay aligned with AWS best practices. Customizations are implemented with AWS CloudFormation templates and service control policies (SCPs).

This CfCT capability is integrated with AWS Control Tower lifecycle events, so that your resource deployments remain synchronized with your landing zone. For example, when a new account is created through account factory, all resources attached to the account are deployed automatically. You can deploy the custom templates and policies to individual accounts and organizational units (OUs) within your organization.

The following video describes best practices for deploying a scalable CfCT pipeline and common CfCT customizations.

The following section provides architectural considerations and configuration steps for deploying Customizations for AWS Control Tower (CfCT). It includes a link to the [AWS CloudFormation](https://aws.amazon.com/cloudformation/) template that launches, configures, and runs the required AWS services, in alignment with AWS best practices for security and availability.
This topic is intended for IT infrastructure architects and developers who have practical experience architecting in the AWS Cloud.

For information about the latest updates and changes to Customizations for AWS Control Tower (CfCT), refer to the CHANGELOG.md file in the GitHub repository.

Architecture overview

Deploying CfCT builds the following environment in the AWS Cloud.

Figure 1: Customizations for AWS Control Tower architecture

CfCT includes an AWS CloudFormation template that you deploy in your AWS Control Tower management account. The template launches all the components necessary to build the workflows, so you can customize your AWS Control Tower landing zone.

Note

CfCT must be deployed in the AWS Control Tower home Region and in the AWS Control Tower management account, because that is where your AWS Control Tower landing zone is deployed. For information about setting up an AWS Control Tower landing zone, refer to Getting started (p. 16).

As you deploy CfCT, it packages and uploads the custom resources to the code pipeline source, by means of Amazon Simple Storage Service (Amazon S3). The upload process automatically invokes the service control policies (SCPs) state machine and the AWS CloudFormation StackSets state machine to deploy the SCPs at the OU level, or to deploy stack instances at the OU or account level.

Note

By default, CfCT creates an Amazon S3 bucket to store the pipeline source, but you can change the location to an AWS CodeCommit repository. For more information, refer to Set up Amazon S3 as the configuration source (p. 68) (p. 68).

CfCT deploys two workflows:

- an AWS CodePipeline workflow
- and an AWS Control Tower lifecycle event workflow.

The AWS CodePipeline workflow

The AWS CodePipeline workflow configures AWS CodePipeline, AWS CodeBuild projects, and AWS Step Functions that orchestrate the management of AWS CloudFormation StackSets and SCPs in your organization.
When you upload the configuration package, CfCT invokes the code pipeline to run three stages.

- **Build Stage** – validates the contents of the configuration package using AWS CodeBuild.
- **SCP Stage** – invokes the service control policy state machine, which calls the AWS Organizations API to create SCPs.
- **AWS CloudFormation Stage** – invokes the stack set state machine to deploy the resources specified in the list of accounts or OUs, which you've provided in the manifest file (p. 71).

At each stage, the code pipeline invokes the stack set and SCP step functions, which deploy custom stack sets and SCPs to the targeted individual accounts, or to an entire organizational unit.

**Note**
For detailed information about customizing the configuration package, refer to CfCT customization guide (p. 69).

**The AWS Control Tower lifecycle event workflow**

When a new account is created in AWS Control Tower, a lifecycle event (p. 1416) can invoke the AWS CodePipeline workflow. You can customize the configuration package through this workflow, which consists of an Amazon EventBridge event rule, an Amazon Simple Queue Service (Amazon SQS) first-in-first-out (FIFO) queue, and an AWS Lambda function.

When the Amazon EventBridge event rule detects a matching lifecycle event, it passes the event to the Amazon SQS FIFO queue, invokes the AWS Lambda function, and invokes the code pipeline to perform downstream deployment of stack sets and SCPs.

**Cost**

The cost for running CfCT depends on the number of AWS CodePipeline runs, the duration of AWS CodeBuild runs, the number and duration of AWS Lambda functions, and the number of Amazon EventBridge events published. For example, if you run 100 builds in one month using build.general1.small where each build runs for five minutes, then the approximate cost for running CfCT is $3.00 per month. For full details, you can review the pricing webpage for each AWS service you are running.

The Amazon Simple Storage Service (Amazon S3) bucket and AWS CodeCommit Git-based repository resources are retained after you delete the template, to protect your configuration information. Depending on the option you select, you are charged based on the amount of data stored in the Amazon S3 bucket and the number of Git requests (not applicable to Amazon S3 resource). Refer to Amazon S3 and AWS CodeCommit pricing for details.

**Component services**

The following AWS services are components of Customizations for AWS Control Tower (CfCT).

**AWS CodeCommit**

Based on your input to the AWS CloudFormation template, CfCT can create an AWS CodeCommit repository with the same sample configuration that's explained in the Amazon Simple Storage Service section.

To clone the CfCT AWS CodeCommit repository to your local computer, you must create credentials that give you temporary access to the repository, as explained in the AWS CodeCommit User Guide. For information about version compatibility, see Setting up for AWS CodeCommit.
AWS CodePipeline

AWS CodePipeline validates, tests, and implements changes based on updates to the configuration package, which you'll make in either the default Amazon S3 bucket or the AWS CodeCommit repository. For more information about changing the configuration source control to AWS CodeCommit, refer to Using Amazon S3 as the Configuration Source (p. 68). The pipeline includes stages to validate and manage the configuration files and templates, core accounts, AWS Organizations service control policies, and AWS CloudFormation StackSets. For more information about the pipeline stages, refer to CfCT customization guide (p. 69).

AWS Key Management Service

CfCT creates an AWS Key Management Service (AWS KMS) CustomControlTowerKMSKey encryption key. This key is used to encrypt objects in the Amazon S3 configuration bucket, Amazon SQS queue, and sensitive parameters in the AWS Systems Manager Parameter Store. By default, only roles provisioned by CfCT have permission to perform encryption or decryption operations with this key. For access to the configuration file, FIFO queue, or Parameter Store SecureString values, administrators must be added to the CustomControlTowerKMSKey policy. Automatic key rotation is enabled by default.

AWS Lambda

CfCT uses AWS Lambda functions to invoke the installation components during the initial installation and deployment of AWS CloudFormation StackSets or AWS Organizations SCPs during an AWS Control Tower lifecycle event.

Amazon Simple Notification Service

CfCT may publish notifications, such as pipeline approval to Amazon Simple Notification Service (Amazon SNS) topics during the workflow. Amazon SNS is launched only when you choose to receive pipeline approval notifications.

Amazon Simple Storage Service

When you deploy CfCT, CfCT creates an Amazon Simple Storage Service (Amazon S3) bucket with a unique name:

Example: Amazon S3 bucket name

custom-control-tower-configuration-accountID-region

The bucket contains a sample configuration file called _custom-control-tower-configuration.zip

Notice the leading underscore in the file name.

This zip file provides a sample manifest and the related sample templates that describe the necessary folder structure. These examples help you develop a configuration package to customize your AWS Control Tower landing zone. The sample manifest identifies the required configurations for stack sets and service control policies (SCPs) you'll need, when you implement your customizations.

You can use this sample configuration package as a model, to develop and upload your custom package, which triggers the CfCT configuration pipeline automatically.

For information about customizing the configuration file, see CfCT customization guide (p. 69).
Amazon Simple Queue Service

CfCT uses an Amazon Simple Queue Service (Amazon SQS) FIFO queue to capture lifecycle events from Amazon EventBridge. It triggers an AWS Lambda function, which invokes AWS CodePipeline to deploy AWS CloudFormation StackSets or SCPs. For more information about SCPs, see [AWS Organizations](#).

AWS Step Functions

CfCT creates Step Functions to orchestrate customization deployments. These Step Functions translate configuration files to deploy the customizations as needed across environments.

AWS Systems Manager Parameter Store

AWS Systems Manager Parameter Store stores the CfCT configuration parameters. These parameters allow you to integrate related configuration templates. For example, you can configure each account to log AWS CloudTrail data to a centralized Amazon S3 bucket. Also, the Systems Manager Parameter Store provides a centralized location where administrators can view CfCT inputs and parameters.

Deployment considerations

Be sure to launch Customizations for AWS Control Tower (CfCT) in the same account and Region where your AWS Control Tower landing zone is deployed; that is, you must deploy it in the AWS Control Tower management account in your AWS Control Tower home Region. By default, CfCT creates and runs the landing zone configuration package by setting up a configuration pipeline in that account and Region.

Prepare for deployment

You have some options when you prepare your AWS CloudFormation template for initial deployment. You can choose the configuration source, and you can allow for manual approval of pipeline deployments. The next two sections explain more about these options.

Choose your configuration source

By default, the template creates an Amazon Simple Storage Service (Amazon S3) bucket to store the sample configuration package as a .zip file called _custom-control-tower-configuration.zip_. The Amazon S3 bucket is version controlled, and you can update the configuration package as needed. For information about updating the configuration package, refer to [Using Amazon S3 as the Configuration Source](#) (p. 68).

**Note**
The sample configuration package filename begins with an underscore (_) so that AWS CodePipeline is not initiated automatically. When you have finished customizing the configuration package, be sure to upload the custom-control-tower-configuration.zip without the underscore (_) in order to begin the deployment in AWS CodePipeline.

You can change the storage location of the configuration package from the S3 bucket to an AWS CodeCommit Git repository by selecting the AWS CodeCommit option in the AWS CloudFormation parameter. This option enables you to manage version control easily.

**Note**
When you're using the default S3 bucket, be sure that the configuration package is available as a .zip file. When you're using the AWS CodeCommit repository, be sure that the configuration package is placed in the repository without zipping the files. For information about creating and storing the configuration package in AWS CodeCommit, see [CfCT customization guide](#) (p. 69).
You can use the sample configuration package to create your own custom configuration source. When you are ready to deploy your custom configurations, manually upload the configuration package, either to the Amazon S3 bucket or to the AWS CodeCommit repository. The pipeline begins automatically when you upload the configuration file.

**Note**
When you're using AWS CodeCommit to store the configuration package, it is not necessary to zip the package. For information about creating and storing the configuration package in AWS CodeCommit, refer to [CfCT customization guide](p. 69).

**Choose your pipeline configuration approval parameters**

The AWS CloudFormation template provides the option to approve the deployment of configuration changes manually. By default, manual approval is not enabled. For more information, refer to [Step 1. Launch the stack](p. 64).

When manual approval is enabled, the configuration pipeline validates the customizations made to the AWS Control Tower file manifest and templates, then it pauses the process until manual approval is granted. After approval, the deployment proceeds to run the remaining pipeline stages, as needed, to implement the Customizations for AWS Control Tower (CfCT) functionality.

You can use the manual approval parameter to keep the customizations for the AWS Control Tower configuration from running, by rejecting the first attempt to run through the pipeline. This parameter also allows you to validate customizations for the AWS Control Tower configuration changes manually, as a final control before implementation.

**To update Customizations for AWS Control Tower**

If you have previously deployed CfCT, you must update the AWS CloudFormation stack to get the latest version of the CfCT framework. For details, refer to [Update the Stack](p. 66).

**Template and source code**

Customizations for AWS Control Tower (CfCT) are deployed in your management account after you launch your AWS CloudFormation template. You can download the template from GitHub and then launch it from AWS CloudFormation.

The `customizations-for-aws-control-tower.template` deploys the following:

- An AWS CodeBuild project
- An AWS CodePipeline project
- An Amazon EventBridge rule
- AWS Lambda functions
- An Amazon Simple Queue Service queue
- An Amazon Simple Storage Service bucket with a sample configuration package
- AWS Step Functions

**Note**
You can customize the template based on your specific requirements.

**Source code repository**

You can visit our [GitHub repository](https://github.com) to download the templates and scripts for CfCT, and to share your landing zone customizations with others.
Automated deployment

Before you launch the automated deployment, review the considerations (p. 62). Follow the step-by-step instructions in this section to configure and deploy the solution into your AWS Control Tower management account.

**Time to deploy:** Approximately 15 minutes

**Prerequisites**

CfCT must be deployed in your AWS Control Tower management account, and in your AWS Control Tower home Region. If you do not have a landing zone set up, see *Getting started* (p. 16).

**Deployment steps**

The procedure for deploying CfCT consists of two major steps. For detailed instructions, follow the links for each step.

**Step 1. Launch the stack (p. 64)**

- Launch the AWS CloudFormation template into your management account.
- Review the template parameters, and adjust if necessary.

**Step 2. Create a custom package (p. 66)**

- Create a custom configuration package.

**Important**

To download the correct AWS CloudFormation template and launch CfCT, follow the GitHub link given in this section. Do not follow older links to any previously specified S3 buckets.

**Step 1. Launch the stack**

The AWS CloudFormation template in this section deploys *Customizations for AWS Control Tower* (CfCT) in your account.

**Note**

You are responsible for the cost of the AWS services used while you run CfCT. For more details, see *Cost* (p. 60).

1. To launch *Customizations for AWS Control Tower*, download the template from GitHub and then launch it from AWS CloudFormation.
2. The template launches in the US East (N. Virginia) Region by default. To launch CfCT in a different AWS Region, use the Region selector in the console navigation bar.

**Note**

CfCT must be launched in the same Region and account where you deployed your AWS Control Tower landing zone, which is your home Region.

3. On the Create stack page, verify that the correct template URL shows in the URL text box and choose Next.
4. On the Specify stack details page, assign a name to your CfCT stack.
5. Under Parameters, review the following parameters and modify them in the template, if necessary.
### Pipeline Configuration

<table>
<thead>
<tr>
<th><strong>Pipeline Approval Stage</strong></th>
<th>No</th>
<th>Choose whether to change the pipeline configuration from the default automated approval stage to a manual approval stage. For more information, see the section called “CfCT customization guide” (p. 69).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pipeline Approval Email Address</strong></td>
<td>&lt;Optional Input&gt;</td>
<td>The email address for approval notifications. To use this parameter, you must set the Pipeline Approval Stage parameter to Yes.</td>
</tr>
<tr>
<td><strong>AWS CodePipeline Source</strong></td>
<td>Amazon S3</td>
<td>The source for AWS CodePipeline to help you select where to store and configure the CfCT customizations.</td>
</tr>
</tbody>
</table>

### AWS CodeCommit Setup

<table>
<thead>
<tr>
<th><strong>Existing CodeCommit Repository?</strong></th>
<th>No</th>
<th>Choose whether to use an existing CodeCommit Git repository. If you choose Yes, you must set the CodePipeline Source parameter to AWS CodeCommit.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CodeCommit Repository Name</strong></td>
<td>custom-control-tower-configuration</td>
<td>The Git repository name. To use this parameter, you must set the AWS CodePipeline Source parameter to AWS CodeCommit. This name is used to create a new Git repository, and must be unique. If you provide the name of an existing Git repository, you must set the Existing CodeCommit Repository? parameter to Yes and enter the exact name of that repository.</td>
</tr>
<tr>
<td><strong>CodeCommit Branch Name</strong></td>
<td>main</td>
<td>The Git branch where the customization package is stored. Git repositories can have many branches. This is the default name given to the branch in the Git repository. To use this parameter, you must set the CodePipeline Source parameter to AWS CodeCommit.</td>
</tr>
</tbody>
</table>
Step 2. Create a custom package

With the launched stack, you can add customizations to your AWS Control Tower landing zone and service control policies (SCPs) by customizing the included configuration package. For detailed instructions on creating a custom package, refer to the CfCT customization guide (p. 69).

**Note**

The pipeline does not run without uploading the custom configuration package.

Update the stack

If you previously deployed Customizations for AWS Control Tower (CfCT), follow the procedure to update the AWS CloudFormation stack for the latest version of the CfCT framework.

**Important**

Before you can complete the following procedure, you must upload the latest template from GitHub to an Amazon Simple Storage Service (Amazon S3) bucket. For instructions on how to get started with Amazon S3, see Getting started with Amazon S3 in the Amazon Simple Storage Service User Guide.

1. Sign in to the AWS CloudFormation console.
2. Select your existing **Customizations for AWS Control Tower** (CfCT) CloudFormation stack, and then select **Update**.

3. Under **Prerequisite — Prepare template**, select **Replace current template**.

4. Under **Specify template**, do the following:
   a. For **Template source**, select **Replace current template**.
   b. For **Amazon S3 URL**, enter the template URL for the template that you previously uploaded from GitGub to Amazon S3, and then choose **Next**.
   c. Verify that the template URL is correct. Then choose **Next** and **Next** again.

5. Under **Parameters**, review the parameters for the template and modify them as necessary. Refer to Step 1. Launch the stack (p. 64) for details about the parameters.

6. Choose **Next**.

7. On the **Configure stack options** page, choose **Next**.

8. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template might create AWS Identity and Access Management (IAM) resources.

9. Choose **View change set** and verify the changes.

10. Choose **Update stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **UPDATE_COMPLETE** in approximately 15 minutes.

---

**Delete a stack set**

You can delete a stack set if you've enabled stack set deletion in the manifest file. By default, the `enable_stack_set_deletion` parameter is set to **false**. In this configuration, no action is taken to delete the associated stack set when a resource is removed from the CfCT manifest file.

If you change the value of `enable_stack_set_deletion` to **true** in the manifest file, CfCT deletes the stack set and all of its resources when you remove an associated resource from the manifest file.

This capability is supported in **v2** of the manifest file.

**Important**

When you initially set the value of `enable_stack_set_deletion` to **true**, the next time you invoke CfCT, **ALL** resources that begin with the prefix `CustomControlTower-`, which have the associated key tag `Key:AWS_Solutions, Value: CustomControlTowerStackSet`, and which are not declared in the manifest file, are staged for deletion.

Here's an example of how to set this parameter in a **manifest.yaml** file:

```yaml
version: 2021-03-15
region: us-east-1
enable_stack_set_deletion: true   #New opt-in functionality

resources:
  - name: demo_resource_1
    resource_file: s3://demo_bucket/resource.template
    deployment_targets:
      accounts:
        - 012345678912
      deploy_method: stack_set
      regions:
        - us-east-1
        - us-west-2
```

67
Set up Amazon S3 as the configuration source

When you set up Customizations for AWS Control Tower, it stores an initial configuration file, called `_custom-control-tower-configuration.zip` file in an Amazon Simple Storage Service (Amazon S3) bucket, named `custom-control-tower-configuration-account-ID-region`.

**Note**

If you choose to download and modify this file, remember to zip the changes, save as a new file named `custom-control-tower-configuration.zip`, and then upload it back to the same Amazon S3 bucket.

The Amazon S3 bucket is the default source of the pipeline. When default settings are in place, uploading a configuration zip file without the underscore prefix in the file name to the S3 bucket will initiate the pipeline automatically.

The zip file is protected by Server-Side Encryption (SSE) with AWS Key Management Service (AWS KMS), and denial of use of the KMS key. For access to the zip file, you must update the KMS Key Policy to specify the role(s) that should be granted access. The role may be an administrator role, a user, or both.

Follow this procedure:

1. Navigate to the AWS Key Management Service console.
2. In Customer Managed Keys, select `CustomControlTowerKMSKey`.
3. Select the Key policy tab. Then, select Edit.
4. In the Edit key policy page, find the Allow Use of the key section in the code, and add one of the following permissions:
   - To add an administration role:
     
     \[
     \text{arn:aws:iam::*<account-ID>:role/<administrator-role>}
     \]
   - To add a user:
     
     \[
     \text{arn:aws:iam::*<account-ID>:user/<username>}
     \]
5. Select Save Changes.
6. Navigate to the Amazon S3 console, find the S3 bucket containing the configuration zip file, and select download.
7. Make the necessary configuration changes to the manifest file and template files. For information about customizing the manifest and template files, see the section called “CfCT customization guide” (p. 69).
8. Upload your changes:
   a. Zip the modified configuration files, and name the file: `custom-control-tower-configuration.zip`.
   b. Upload the file to Amazon S3 using SSE with the AWS KMS master-key: `CustomControlTowerKMSKey`.

- name: demo_resource_2
  resource_file: s3://demo_bucket/resource.template
  deployment_targets:
    accounts:
      - 012345678912
    deploy_method: stack_set
  regions:
    - us-east-1
    - eu-north-1
Collection of operational metrics

Customizations for AWS Control Tower (CfCT) includes an option to send anonymous operational metrics to AWS. AWS uses this data to understand how customers are using CfCT, as well as other related services and products. When data collection is enabled, the following information is sent to AWS:

- **Solution ID:** The AWS solution identifier
- **Unique ID (UUID):** Randomly generated, unique identifier for each deployment
- **Timestamp:** Data-collection timestamp
- **State Machine Execution Count:** Incrementally counts the number of times this state machine runs
- **Manifest Version:** The manifest version used in the configuration

**Note**
AWS owns the data it collects. Data collection is subject to the AWS Privacy Policy.

To opt out of sending anonymous operational metrics to AWS, complete one of the following tasks:

- **Update the AWS CloudFormation template mapping section as follows:**

  from

  ```
  AnonymousData:
  SendAnonymousData:
    Data: Yes
  ```

  to

  ```
  AnonymousData:
  SendAnonymousData:
    Data: No
  ```

- **After CfCT is deployed, find the /org/primary/metrics_flag SSM parameter key in the Parameter Store console, and update the value to No.**

CfCT customization guide

The Customizations for AWS Control Tower (CfCT) guide is for administrators, DevOps professionals, independent software vendors, IT infrastructure architects, and systems integrators who want to customize and extend their AWS Control Tower environments for their company and customers. It provides information about customizing and extending the AWS Control Tower environment with the CfCT customization package.

**Note**
To deploy and configure (CfCT), you must deploy and process a configuration package through AWS CodePipeline. The following sections describe the process in detail.

Code pipeline overview

The configuration package requires Amazon Simple Storage Service (Amazon S3) and AWS CodePipeline. The configuration package contains these items:

- A manifest file
- An accompanying set of templates
• Other JSON files for describing and implementing your AWS Control Tower environment customizations

By default, the _custom-control-tower-configuration.zip configuration package is loaded in an Amazon S3 bucket with the following naming convention:

\textit{custom-control-tower-configuration-\texttt{accountID-region}}.

\textbf{Note}

By default, CfCT creates an Amazon S3 bucket to store the pipeline source, but you can change the source location to an AWS CodeCommit repository. For more information, see \textit{Edit a pipeline in CodePipeline} in the \textit{AWS CodePipeline User Guide}.

The \textit{manifest file} is a text file that describes the AWS resources you can deploy to customize your landing zone. CodePipeline does these tasks:

• extracts the manifest file, accompanying set of templates, and other JSON files
• performs manifest and template validations
• invokes sections in the manifest file to run specific \textit{pipeline stages} (p. 70).

When you update the configuration package by customizing the manifest file and removing the underscore (_) from the configuration package filename, it automatically initiates AWS CodePipeline.

\textbf{Note}

The sample configuration package filename begins with an underscore (_) so that AWS CodePipeline is not automatically triggered. When you have completed the customization of the configuration package, upload the file \texttt{custom-control-tower-configuration.zip} without the underscore (_) in order to trigger the deployment in AWS CodePipeline.

\section*{AWS CodePipeline stages}

The CfCT pipeline requires several AWS CodePipeline stages to implement and update your AWS Control Tower environment.

\textbf{1. Source stage}

The source stage is the initial stage. Your customized configuration package initiates this pipeline stage. The source for the AWS CodePipeline can be either an Amazon S3 bucket or an AWS CodeCommit repository, in which the configuration package can be hosted.

\textbf{2. Build stage}

The build stage requires AWS CodeBuild to validate the contents of the configuration package. These checks include testing the \texttt{manifest.yaml} file syntax and schema, along with all AWS CloudFormation templates included in the package or remotely hosted, using AWS CloudFormation \texttt{validate-template} and \texttt{cfn_nag}. If the manifest file and AWS CloudFormation templates pass the tests, the pipeline continues to the next stage. If the tests fail, you can review the CodeBuild logs to identify the issue and edit the configuration source file as needed.

\textbf{3. Manual approval stage (optional)}

The manual approval stage is optional. If you enable this stage, it provides additional control over the configuration pipeline. It pauses the pipeline during deployment, until an approval is given. You can opt into manual approval by editing the \textbf{Pipeline Approval Stage} parameter to \textbf{Yes} when you launch the stack.

\textbf{4. Service control policy stage}

The service control policy stage invokes the service control policy state machine to call AWS Organizations APIs that create service control policies (SCPs).
5. **AWS CloudFormation resource stage**

The AWS CloudFormation resource stage invokes the stack set state machine to deploy the resources specified in the list of accounts or organizational units (OUs), which you provided in the manifest file. The state machine creates the AWS CloudFormation resources in the order that they are specified in the manifest file, unless a resource dependency is specified.

## Define a custom configuration

You'll define your custom AWS Control Tower configuration with the manifest file, the accompanying set of templates, and other JSON files. You'll package these files into a folder structure and place them in the Amazon S3 bucket as a `.zip` file, as shown in the following code example.

### Custom configuration folder structure

- manifest.yaml
- policies/ [optional]
  - service control policies files (*.json)
- templates/ [optional]
  - template files for AWS CloudFormation Resources (*.template)

The previous example depicts the structure of a custom configuration folder. The folder structure stays the same whether you choose Amazon S3 or an AWS CodeCommit repository as your source storage location. If you choose Amazon S3 as source storage, compress all the folders and files into a `custom-control-tower-configuration.zip` file, and upload only the `.zip` file to the designated Amazon S3 bucket.

**Note**

If you are using AWS CodeCommit, place the files in the repository without zipping the files.

### The manifest file

The manifest `.yaml` file is a text file that describes your AWS resources. The following example shows the structure of the manifest file.

```yaml
---
region: String
version: 2021-03-15
resources:
  # set of CloudFormation resources or SCP policies
...
```

As shown in the previous code example, the first two lines of the manifest file specify the values of the `region` and the `version` keywords. Here are the definitions of those keywords.

**region** – A text string for the AWS Control Tower default Region. This value must be a valid AWS Region name (such as us-east-1, eu-west-1, or ap-southeast-1). The AWS Control Tower home Region is the default when you create custom AWS Control Tower resources (such as AWS CloudFormation StackSets), unless a more resource-specific Region is specified.

```yaml
region: your-home-region
```

**version** – The manifest schema version number. The latest supported version is 2021-03-15.

```yaml
version: 2021-03-15
```
Note
We strongly recommend you use the latest version. To update manifest properties in the latest version, refer to [Manifest version upgrades](p. 81).

The next keyword shown in the previous example is the resources keyword. The resources section of the manifest file is highly structured. It contains a detailed list of AWS resources, which will be deployed automatically by the CfCT pipeline. These descriptions of resources and their available parameters are given in the next section.

The resources section of the manifest file

This topic describes the resources section of the manifest file, where you'll define the resources that are required for your customizations. This section of the manifest file begins at the keyword resources and continues to the end of the file.

The resources section of the manifest file specifies the AWS CloudFormation StackSets or AWS Organizations SCPs, which CfCT deploys automatically through the code pipeline. You can list OUs, accounts, and Regions to deploy stack instances.

Stack instances are deployed at the account level instead of the OU level. SCPs are deployed at the OU level. For more information, see [Build your own customizations](#).

The following example template describes the possible entries that are available for the resources section of the manifest file.

```yaml
resources: # List of resources
  - name: [String]
    resource_file: [String] [Local File Path, S3 URI, S3 URL]
    deployment_targets: # account and/or organizational unit names
      accounts: # array of strings, [0-9]{12}
        - 012345678912
        - AccountName1
      organizational_units: # array of strings
        - OuName1
        - OuName2
    deploy_method: scp | stack_set
    parameters: # List of parameters [SSM, Alfred, Values]
      - parameter_key: [String]
        parameter_value: [String]
    export_outputs: # list of ssm parameters to store output values
      - name: /org/member/test-ssm/app-id
        value: $[output_ApplicationId]
    regions: # list of strings
      - [String]
```

The remainder of this topic gives detailed definitions for the keywords shown in the previous code example.

**name** – The name that is associated with the AWS CloudFormation StackSets. The string you provide assigns a more user-friendly name for a stack set.

- **Type:** String
- **Required:** Yes
- **Valid Values:** a-z, A-Z, 0-9, and an underscore (_). Any other character is automatically replaced with an underscore (_).

**description** – The description for the resource.

- **Type:** String
• **Required:** No

**resource_file** – This file can be specified as the relative location to the manifest file, an Amazon S3 URI or URL that points to an AWS CloudFormation template or AWS Organizations service control policy in JSON for creating AWS CloudFormation resources or SCPs.

• **Type:** String
• **Required:** Yes

1. The following example shows the `resource_file`, given as a relative location to the resource file inside the configuration package.

```yaml
resources:
  - name: SecurityRoles
    resource_file: templates/custom-security.template
```

2. The following example shows the resource file given as an Amazon S3 URI

```yaml
resources:
  - name: SecurityRoles
    resource_file: s3://bucket-name/[key-name]
```

3. The following example shows the resource file given as an Amazon S3 HTTPS URL

```yaml
resources:
  - name: SecurityRoles
    resource_file: https://bucket-name.s3.Region.amazonaws.com/key-name
```

**Note**
If you provide an Amazon S3 URL, verify that the bucket policy allows read access for the AWS Control Tower management account from which you are deploying CfCT. If you provide an Amazon S3 HTTPS URL, verify that the path uses dot notation. For example, S3.us-west-1. CfCT does not support endpoints that contain a dash between S3 and the Region, such as S3-us-west-2.

4. The following example shows an Amazon S3 bucket policy and an ARN where resources are stored.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {"AWS": "arn:aws:iam::AccountId:root"},
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::my-bucket/*"
        }
    ]
}
```

You'll replace the `AccountId` variable shown in the example with the AWS account ID for the management account that is deploying CfCT. For more examples, refer to [Bucket policy examples](#) in the Amazon Simple Storage Service User Guide.

**parameters** – Specifies the name and value for AWS CloudFormation parameters.

• **Type:** MapList
• Required: No

The parameters section contains pairs of key/value parameters. The following pseudo template outlines the parameters section.

```yaml
parameters:
  - parameter_key: [String]
    parameter_value: [String]
```

- **parameter_key** – The key associated with the parameter.
  - **Type:** String
  - **Required:** Yes (under parameters property)
  - **Valid Values:** a-z, A-Z, and 0-9

- **parameter_value** – The input value associated with the parameter.
  - **Type:** String
  - **Required:** Yes (under parameters property)

- **deploy_method** – The deployment method for deploying resource(s) into the account. Currently, deploy_method supports deploying resources using the stack_set option for resource deployment through AWS CloudFormation StackSets, or the scp option if you are deploying SCPs.
  - **Type:** String
  - **Valid Values:** stack_set | scp
  - **Required:** Yes

- **deployment_targets** – List of accounts or Organizational Units (OUs), into which CfCT will deploy the AWS CloudFormation resources, specified as accounts or organizational_units.
  - **Note**
    If you want to deploy an SCP, the target must be an OU, not an account.
  - **Type:** List of string account name or account number to indicate that this resource will be deployed into the given account list, or OU names to indicate that this resource will be deployed into the given OU list.
  - **Required:** At least one of accounts or organizational_units
    - **accounts:**
      - **Type:** List of string account name or account number to indicate that this resource will be deployed into the given account list.
    - **organizational_units:**
      - **Type:** List of string OU names to indicate that this resource will be deployed into a given OU list. If you provide an OU that doesn't contain accounts and the accounts property is not added, CfCT only creates the stack set.
      - **Note**
        The organization's management account ID is not an allowed value. CfCT does not support deploying stack instances into the organization's management account.

- **export_outputs** – List of name/value pairs that denote SSM parameter keys. These SSM parameter keys allow you to store template outputs into the SSM parameter store. The output is intended for reference by other resources, defined earlier in the manifest file.
export_outputs: # List of SSM parameters
- name: [String]
  value: [String]

- **Type:** List of name and value key pairs. The name contains the name string of an SSM parameter store key, and value contains the parameter's value string.

- **Valid Values:** Any string or the $[output_CfnOutput-Logical-ID]$ variable where CfnOutput-Logical-ID corresponds to the template output variable. For more information about the Outputs section in an AWS CloudFormation template, see Outputs in the AWS CloudFormation User Guide.

- **Required:** No

For example, the following code snippet stores the template VPCID output variable into the SSM parameter key that's named /org/member/audit/vpc_id.

export_outputs: # List of SSM parameters
- name: /org/member/audit/VPC-ID
  value: $[output_VPCID]

**Note**
The export_outputs key name may contain a value other than output. For example, if the name is /org/environment-name, the value may be production.

**regions** – List of Regions in which CfCT will deploy the AWS CloudFormation stack instances.

- **Type:** Any list of AWS commercial Region names, to indicate that this resource will be deployed into the given Region list. If this keyword does not exist in the manifest file, the resources are deployed in the home Region only.

- **Required:** No

### Root OU

CfCT supports **Root** as a value for an organizational unit (OU) under organizational_units in manifest V2 version (2021-03-15).

- If you choose the deployment method of scp, when you add Root under organizational_units, AWS Control Tower applies the policies to all of the OUs under the Root. If you choose the deployment method of stack_set, when you add Root under organizational_units, CfCT deploys the stack sets in all the accounts under the Root that are enrolled in AWS Control Tower, except for the management account.

- As per AWS Control Tower best practices, the management account is intended only to manage member accounts and for billing purposes. Do not run production workloads in the AWS Control Tower management account.

In accordance with best practices guidance, AWS Control Tower deployment puts the management account under the Root OU, so that it has full access and does not run additional resources. For this reason, the **AWSControlTowerExecution** role is not deployed to the management account.

- We recommend that you follow these best practices for the management account. If you have a specific use case that requires you to deploy stacksets in the management account, include accounts as a deployment target and specify the management account. Otherwise, do not include accounts as a deployment target. You must create the missing resources, including required IAM roles, in the management account.
To deploy stacksets in the management account, include accounts as a deployment target and specify the management account. Otherwise, do not include accounts as a deployment target.

```yaml
---
region: your-home-region
version: 2021-03-15
resources:
  ...truncated...
  deployment_targets:
    organizational_units:
      - Root
---
```

**Note**
The Root OU feature is supported only in the V2 version of the manifest file (2021-03-15). If you add **Root** as an OU under organizational_units, do not add any other OUs.

### Nested OU

CfCT supports listing one or more nested OUs under the organizational_units keyword in manifest V2 version (2021-03-15).

A complete path (excluding Root) for the nested OU is required, using a colon as the separator between OUs. For deployment method **scp**, AWS Control Tower deploys the SCPs to the last OU in the nested OU path. For deployment method **stack_set**, AWS Control Tower deploys the stack sets to all the accounts under the last OU in the nested OU path.

For example, consider the path `OUname1:OUname2:OUname3`. The last OU in the path is `OUname3`. CfCT deploys the SCPs to `OUname3` and stack sets to all of the accounts directly under `OUname3`, only.

```yaml
---
region: your-home-region
version: 2021-03-15
resources:
  ...truncated...
  deployment_targets:
    organizational_units:
      - OUname1:OUname2:OUname3
---
```

**Note**
The nested OU feature is supported only in the V2 version of the manifest file (2021-03-15).

### Build your own customizations

To build your own customizations, you can modify the `manifest.yaml` file by adding or updating service control policies (SCPs) and AWS CloudFormation resources. For resources that must be deployed, you can add or remove accounts and OUs. You can add or modify the templates in the package folders, create your own folders, and reference the templates or folders in the `manifest.yaml` file.

This section explains the two main parts of building your own customizations:

- how to set up your own configuration package for service control policies
• how to set up your own configuration package for AWS CloudFormation stack sets

Set up a configuration package for service control policies

This section explains how to create a configuration package for service control policies (SCPs). The two main parts of this process are (1) prepare the manifest file, and (2) prepare your folder structure.

Step 1: Edit the manifest.yaml file

Use the sample manifest.yaml file as your starting point. Enter all necessary configurations. Add the resource_file and deployment_targets details.

The following snippet shows the default manifest file.

```yaml
---
region: us-east-1
version: 2021-03-15
resources: []
```

The value for region is added automatically during deployment. It must match the Region where you deployed CfCT. This Region must be the same as the AWS Control Tower region.

To add a custom SCP in the example-configuration folder in the zip package stored in the Amazon S3 bucket, open the example-manifest.yaml file and begin editing.

```yaml
---
region: your-home-region
version: 2021-03-15
resources:
- name: test-preventive-controls
description: To prevent from deleting or disabling resources in member accounts
resource_file: policies/preventive-controls.json
deploy_method: scp
#Apply to the following OU(s)
deployment_targets:
  organizational_units: #array of strings
    - OUName1
    - OUName2
```

The following snippet shows an example of a customized manifest file. You can add more than one policy in a single change.

```yaml
---
region: us-east-1
version: 2021-03-15
resources:
- name: block-s3-public-access
description: To S3 buckets to have public access
resource_file: policies/block-s3-public.json
deploy_method: scp
#Apply to the following OU(s)
deployment_targets:
  organizational_units: #array of strings
    - OUName1
    - OUName2
```
Step 2: Create a folder structure

You can skip this step if you are using an Amazon S3 URL for the resource file and using parameters with key/value pairs.

You must include an SCP policy in JSON format to support the manifest, because the manifest file references the JSON file. Ensure that the file paths match the path information provided in the manifest file.

- A policy JSON file contains the SCPs to be deployed to OUs.

The following snippet shows the folder structure for the sample manifest file.

```
- manifest.yaml
- policies/
  - block-s3-public.json
```

The following snippet is an example of a block-s3-public.json policy file.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GuardPutAccountPublicAccessBlock",
      "Effect": "Deny",
      "Action": "s3:PutAccountPublicAccessBlock",
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

Set up a configuration package for AWS CloudFormation StackSets

This section explains how to set up a configuration package for AWS CloudFormation StackSets. The two main parts of this process are: (1) prepare the manifest file, and (2) update the folder structure.

Step 1: Edit the existing manifest file

Add the new AWS CloudFormation StackSets information to the manifest file that you previously edited.

Just for review, the following snippet contains the same customized manifest file that was shown previously to set up a configuration package for SCPs. Now you can edit this file further, to include the details about your resources.

```
---
region: us-east-1
version: 2021-03-15

resources:
  - name: block-s3-public-access
description: To S3 buckets to have public access
resource_file: policies/block-s3-public.json
deploy_method: scp
  #Apply to the following OU(s)
```
deployment_targets:
organizational_units: #array of strings
- OUName1
- OUName2

The following snippet shows an edited sample manifest file that contains the resources details. The order of resources determines the execution order for creating resources dependencies. You can edit the following example manifest file according to your business requirements.

```yaml
---
region: your-home-region
version: 2021-03-15
...truncated...
resources:
- name: stackset-1
  resource_file: templates/create-ssm-parameter-keys-1.template
  parameters:
    - parameter_key: parameter-1
      parameter_value: value-1
  deploy_method: stack_set
  deployment_targets:
    accounts: # array of strings, [0-9]{12}
    - account number or account name
    - 123456789123
    organizational_units: #array of strings, ou ids, ou-xxxx
    - OuName1
    - OUName2
  export_outputs:
    - name: /org/member/test-ssm/app-id
      value: $[output_ApplicationId]
  regions:
    - region-name

- name: stackset-2
  resource_file: s3://bucket-name/key-name
  parameters:
    - parameter_key: parameter-1
      parameter_value: value-1
  deploy_method: stack_set
  deployment_targets:
    accounts: # array of strings, [0-9]{12}
    - account number or account name
    - 123456789123
    organizational_units: #array of strings
    - OuName1
    - OUName2
  regions:
    - region-name
```

The following example shows that you can add more than one AWS CloudFormation resource in the manifest file.

```yaml
---
region: us-east-1
version: 2021-03-15
resources:
- name: block-s3-public-access
  description: To S3 buckets to have public access
  resource_file: policies/block-s3-public.json
  deploy_method: scp
```
# Apply to the following OU(s)

deployment_targets:
  organizational_units: #array of strings
  - Custom
  - Sandbox

- name: transit-network
  resource_file: templates/transit-gateway.template
  parameter_file: parameters/transit-gateway.json
  deploy_method: stack_set
  deployment_targets:
    accounts: # array of strings, [0-9]{12}
      - Prod
      - 123456789123 #Network
    organizational_units: #array of strings
      - Custom
  export_outputs:
    - name: /org/network/transit-gateway-id
      value: $[output_TransitGatewayID]
  regions:
    - us-east-1

## Step 2: Update the folder structure

When you update the folder structure, you can include all supporting AWS CloudFormation template files and SCP policy files that are in the manifest file. Verify that the file paths match what is provided in the manifest file.

- A template file contains the AWS resources to be deployed in OUs and accounts.
- A policy file contains the input parameters used in the template file.

The following example shows the folder structure for the sample manifest file created in Step 1 (p. 78).

- manifest.yaml
- policies/
  - block-s3-public.json
- templates/
  - transit-gateway.template

### The ‘alfred’ helper and the AWS CloudFormation parameter files

CfCT provides you with a mechanism known as the alfred helper to get the value for an SSM Parameter Store key that’s defined in the AWS CloudFormation template. Using the alfred helper, you can use values that are stored in the SSM Parameter Store and without updating the AWS CloudFormation template. For more information, see What is an AWS CloudFormation template? in the AWS CloudFormation User Guide.

**Important**

The alfred helper has two limitations. Parameters are available only in the home region of the AWS Control Tower management account. As a best practice, consider working with values that don't change from stack instance to stack instance. When the 'alfred' helper retrieves parameters, it chooses a random stack instance from the stack set that exports the variable.

#### Example

Suppose that you have two AWS CloudFormation stack sets. Stack set 1 has one stack instance and deploys to one account in one Region. It creates an Amazon VPC and subnets in an availability zone, and the VPC ID and subnet ID must be passed into stack set 2 as parameter values. Before the VPC ID
and subnet ID can be passed to stack set 2, the VPC ID and subnet ID must be stored in stack set 1 using AWS::SSM::Parameter. For more information, see AWS::SSM::Parameter in the AWS CloudFormation User Guide.

**AWS CloudFormation stack set 1:**

In the following snippet, the alfred helper can gets value for the VPC ID and subnet ID from the parameter store and pass them as input to the StackSet state machine.

```yaml
VpcIdParameter:
  Type: AWS::SSM::Parameter
  Properties:
    Name: '/stack_1/vpc/id'
    Description: Contains the VPC id
    Type: String
    Value: !Ref MyVpc

SubnetIdParameter:
  Type: AWS::SSM::Parameter
  Properties:
    Name: '/stack_1/subnet/id'
    Description: Contains the subnet id
    Type: String
    Value: !Ref MySubnet
```

**AWS CloudFormation stack set 2:**

The snippet shows the parameters that are specified in the AWS CloudFormation stack 2 manifest.yaml file.

```yaml
parameters:
  - parameter_key: VpcId
    parameter_value: $[alfred_ssm_/stack_1/vpc/id]
  - parameter_key: SubnetId
    parameter_value: $[alfred_ssm_/stack_1/subnet/id]
```

**AWS CloudFormation stack set 2.1:**

The snippet shows that you can list alfred_ssm properties to support parameters of type CommaDelimitedList. For more information, see Parameters in the AWS CloudFormation User Guide.

```yaml
parameters:
  - parameter_key: VpcId # Type: String
    parameter_value: $[alfred_ssm_/stack_1/vpc/id']
  - parameter_key: SubnetId # Type: String
    parameter_value: $[alfred_ssm_/stack_1/subnet/id']
  - parameter_key: AvailabilityZones # Type: CommaDelimitedList
    parameter_value:
      - "$[alfred_ssm_/availability_zone_1]"
      - "$[alfred_ssm_/availability_zone_2]"
```

**JSON schema for the customization package**

The JSON schema for the customization package for CfCT is located in the source code repository on GitHub. You can use the schema with many of your favorite development tools, and you may find it helpful for reducing errors when you build your own manifest.yaml file.

**Manifest version upgrades**

For information about the latest version of Customizations for AWS Control Tower (CfCT), see the CHANGELOG.md file in the GitHub repository.
Warning
Version 2.2.0 of Customizations for AWS Control Tower (CfCT) introduced a manifest schema (version 2021-03-15) to align with related AWS service APIs. The manifest schema allows a single manifest.yaml file to manages supported resources (AWS CloudFormation templates and SCPs) through decoupled DevOps workflows.

We strongly recommend that you update the manifest schema from version 2020-01-01 to version 2021-03-15 or later.

CfCT continues to support version 2021-03-15 and 2020-01-01 of the manifest.yaml file. No changes to your existing configuration are required. However, version 2020-01-01 is at End of Support. We no longer provide updates or add enhancements to version 2020-01-01. The Root OU and nested OU features aren't supported in version 2020-01-01.

Deprecated properties in manifest version 2021-03-15:

- organization_policies
- policy_file
- apply_to_accounts_in_ou
- cloudformation_resources
- template_file
- deploy_to_account
- deploy_to_ou
- ssm_parameters

Mandatory upgrade steps

When you upgrade to the manifest schema version 2021-03-15 version, here are the changes you must make to update your files. The next sections outline mandatory and recommended changes for the transition.

Organizations polices

1. Move the SCPs under organization_policies under new property resources.
2. Change the policy_file property to new property resource_file.
3. Change the apply_to_accounts_in_ou to new property deployment_targets. The OU list should be defined under sub-property organizational_units. The accounts sub-property is not supported for organizations policies.
4. Add a new property deploy_method with the value scp.

AWS CloudFormation resources

1. Move the CloudFormation resources under cloudformation_resources under new property resources.
2. Change the template_file property to new property resource_file.
3. Change the deploy_to_ou to new property deployment_targets. The OU list should be defined under sub-property organizational_units.
4. Change the deploy_to_accounts to new property deployment_targets. The account list should be defined under sub-property accounts.
5. Change the ssm_parameters property to new property export_outputs.

Highly recommended upgrade steps

AWS CloudFormation parameters

1. Change the parameter_file property to new property parameters.
2. Remove the file path in the value of the `parameter_file` property.
3. Copy the parameter key and parameter value from the existing parameter JSON file into the new format for the `parameters` property. This would help you manage them in the manifest file.

**Note**
The `parameter_file` property is supported in manifest version 2021-03-15.
Networking in AWS Control Tower

AWS Control Tower provides basic support for networking through VPCs.

If the default configuration or capabilities of the AWS Control Tower VPC do not meet your needs, you can use other AWS services to configure your VPC. For more information about how to work with VPCs and AWS Control Tower, see Building a Scalable and Secure Multi-VPC AWS Network Infrastructure.

Related topics

- For information about how AWS Control Tower works when you enroll accounts that have existing VPCs, see Enrolling existing accounts with VPCs (p. 109).
- With Account Factory, you can provision accounts that include an AWS Control Tower VPC, or you can provision accounts without a VPC. For information about how to delete the AWS Control Tower VPC or configure AWS Control Tower accounts without a VPC, see Walkthrough: Configure AWS Control Tower Without a VPC (p. 1432).
- For information about how to change account settings for VPCs, see the Account Factory documentation on updating an account.
- For more information about working with networking and VPCs in AWS Control Tower, see the section about Networking on the Related information page of this User Guide.

VPCs and AWS Regions in AWS Control Tower

As a standard part of account creation, AWS creates an AWS-default VPC in every Region, even the Regions you are not governing with AWS Control Tower. This default VPC is not the same as a VPC that AWS Control Tower creates for a provisioned account, but the AWS default VPC in a non-governed Region may be accessible to IAM users.

Administrators can enable the Region deny control, so that your end-users do not have permission to connect to a VPC in a Region that's supported by AWS Control Tower but outside your governed Regions. To configure the Region deny control, go to the Landing zone settings page and select Modify settings.

The Region deny control blocks API calls to most services in non-governed Regions. For more information, see Deny access to AWS based on the requested AWS Region (p. 1344).

**Note**

The Region deny control may not prevent IAM users from connecting to an AWS default VPC in a Region where AWS Control Tower is not supported.

Optionally, you can remove the AWS default VPCs in non-governed Regions. To list the default VPC in a Region you can use a CLI command similar to this example:

```bash
aws ec2 --region us-west-1 describe-vpcs --filter Name=isDefault,Values=true
```

Overview of AWS Control Tower and VPCs

Here are some essential facts about AWS Control Tower VPCs:

- The VPC created by AWS Control Tower when you provision an account in Account Factory is not the same as the AWS default VPC.
• When AWS Control Tower sets up a new account in a supported AWS Region, AWS Control Tower automatically deletes the default AWS VPC, and it sets up a new VPC configured by AWS Control Tower.

• Each AWS Control Tower account is allowed one VPC that's created by AWS Control Tower. An account can have additional AWS VPCs within the account limit.

• Every AWS Control Tower VPC has three Availability Zones in all Regions except for the US West (N. California) Region, us-west-1, and two Availability Zones in us-west-1. By default, each Availability Zone is assigned one public subnet and two private subnets. Therefore, in Regions except US West (N. California) each AWS Control Tower VPC contains nine subnets by default, divided across three Availability Zones. In US West (N. California), six subnets are divided across two Availability Zones.

• Each of the subnets in your AWS Control Tower VPC is assigned a unique range, of equal size.

• The number of subnets in a VPC is configurable. For more information about how to change your VPC subnet configuration, see the Account Factory topic.

• Because the IP addresses do not overlap, the six or nine subnets within your AWS Control Tower VPC can communicate with each other in an unrestricted manner.

When working with VPCs, AWS Control Tower makes no distinction at the Region level. Every subnet is allocated from the exact CIDR range that you specify. The VPC subnets can exist in any Region.

Notes

Manage VPC costs
If you set the Account Factory VPC configuration so that public subnets are enabled when provisioning a new account, Account Factory configures VPC to create a NAT Gateway. You will be billed for your usage by Amazon VPC.

VPC and control settings
If you provision Account Factory accounts with VPC internet access settings enabled, that Account Factory setting overrides the control Disallow internet access for an Amazon VPC instance managed by a customer (p. 1346). To avoid enabling internet access for newly provisioned accounts, you must change the setting in Account Factory. For more information, see Walkthrough: Configure AWS Control Tower Without a VPC (p. 1432).

CIDR and Peering for VPC and AWS Control Tower

This section is intended primarily for network administrators. Your company's network administrator usually is the person who selects the overall CIDR range for your AWS Control Tower organization. The network administrator then allocates subnets from within that range for specific purposes.

When you choose a CIDR range for your VPC, AWS Control Tower validates the IP address ranges according to the RFC 1918 specification. Account Factory allows a CIDR block of up to /16 in the ranges of:

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16
- 100.64.0.0/10 (only if your internet provider allows usage of this range)

The /16 delimiter allows up to 65,536 distinct IP addresses.

You can assign any valid IP addresses from the following ranges:

- 10.0.x.x to 10.255.x.x
- 172.16.x.x – 172.31.x.x
• 192.168.0.0 – 192.168.255.255 (no IPs outside of 192.168 range)

If the range you specify is outside of these, AWS Control Tower provides an error message.

The default CIDR range is 172.31.0.0/16.

When AWS Control Tower creates a VPC using the CIDR range you select, it assigns the identical CIDR range to every VPC for every account you create within the organizational unit (OU). Due to the default overlap of IP addresses, this implementation does not initially permit peering among any of your AWS Control Tower VPCs in the OU.

Subnets

Within each VPC, AWS Control Tower divides your specified CIDR range evenly into nine subnets (except in US West (N. California), where it is six subnets). None of the subnets within a VPC overlap. Therefore, they all can communicate with each other, within the VPC.

In summary, by default, subnet communication within the VPC is unrestricted. The best practice for controlling communication among your VPC subnets, if needed, is to set up access control lists with rules that define the permitted traffic flow. Use security groups for control of traffic among specific instances. For more information about setting up security groups and firewalls in AWS Control Tower, see Walkthrough: Set Up Security Groups in AWS Control Tower With AWS Firewall Manager (p. 1434).

Peering

AWS Control Tower does not restrict VPC-to-VPC peering for communication across multiple VPCs. However, by default, all AWS Control Tower VPCs have the same default CIDR range. To support peering, you can modify the CIDR range in the settings of Account Factory so that the IP addresses do not overlap.

If you change the CIDR range in the settings of Account Factory, all new accounts that are subsequently created by AWS Control Tower (using Account Factory) are assigned the new CIDR range. The old accounts are not updated. For example, you can create an account, then change the CIDR range and create a new account, and the VPCs allocated to those two accounts can be peered. Peering is possible because their IP address ranges are not identical.
How AWS Control Tower works with roles to create and manage accounts

In general, roles are a part of identity and access management (IAM) in AWS. For general information about IAM and roles in AWS, see the IAM roles topic in the AWS IAM User Guide.

For specific information about the roles required to use the AWS Control Tower console, see Permissions Required to Use the AWS Control Tower Console (p. 1397).

Roles and account creation

AWS Control Tower creates a customer's account by calling the CreateAccount API of AWS Organizations. When AWS Organizations creates this account, it creates a role within that account, which AWS Control Tower names by passing in a parameter to the API. The name of the role is AWSControlTowerExecution.

AWS Control Tower takes over the AWSControlTowerExecution role for all accounts created by Account Factory. Using this role, AWS Control Tower baselines the account and applies mandatory (and any other enabled) controls, which results in creation of other roles. These roles in turn are used by other services, such as AWS Config.

Note
To baseline an account is to set up its resources, which include Account Factory templates, sometimes referred to as blueprints, and controls. The baselining process also sets up the centralized logging and security audit roles on the account, as part of deploying the templates. AWS Control Tower baselines are contained in the roles that you apply to every enrolled account.

For more information about accounts and resources, see About AWS accounts in AWS Control Tower (p. 105).

The AWSControlTowerExecution role, explained

The AWSControlTowerExecution role must be present in all enrolled accounts. It allows AWS Control Tower to manage your individual accounts and report information about them to your Audit and Log Archive accounts.

The AWSControlTowerExecution role can be added into an account in several ways, as follows:

- For accounts in the Security OU (sometimes called core accounts), AWS Control Tower creates the role at the time of initial AWS Control Tower setup.
- For an Account Factory account created through the AWS Control Tower console, AWS Control Tower creates this role at the time of account creation.
- For a single account enrollment, we ask customers to manually create the role and then enroll the account in AWS Control Tower.
- When extending governance to an OU, AWS Control Tower uses the StackSet-AWSControlTowerExecutionRole to create the role in all accounts in that OU.

Purpose of the AWSControlTowerExecution role:
• AWSControlTowerExecution allows you to create and enroll accounts, automatically, with scripts and Lambda functions.
• AWSControlTowerExecution helps you configure your organization's logging, so that all the logs for every account are sent to the logging account.
• AWSControlTowerExecution allows you to enroll an individual account in AWS Control Tower. First, you must add the AWSControlTowerExecution role to that account. For steps on how to add the role, see Manually add the required IAM role to an existing AWS account and enroll it (p. 113).

Optional conditions for your role trust relationships

You can impose conditions in your role trust policies, to restrict the accounts and resources that interact with certain roles in AWS Control Tower. We strongly recommend that you restrict access to the AWSControlTowerAdmin role, because it allows wide access permissions.

To help prevent an attacker from gaining access to your resources, manually edit your AWS Control Tower trust policy to add at least one aws:SourceArn or aws:SourceAccount conditional to the policy statement. As a security best practice, we strongly recommend adding the aws:SourceArn condition, because it is more specific than aws:SourceAccount, limiting access to a specific account and a specific resource.

If you don't know the full ARN of the resource, or if you are specifying multiple resources, you can use the aws:SourceArn condition with wildcards (*) for the unknown portions of the ARN. For example, arn:aws:controltower:*:123456789012:* works if you don't wish to specify a Region.

The following example demonstrates the use of the aws:SourceArn IAM condition with your IAM role trust policies. Add the condition in your trust relationship for the AWSControlTowerAdmin role, because the AWS Control Tower service principal interacts with it.

As shown in the example, the source ARN is of the format:
arn:aws:controltower:${HOME_REGION}:${CUSTOMER_AWSACCOUNT_id}:*

Replace the strings ${HOME_REGION} and ${CUSTOMER_AWSACCOUNT_id} with your own home Region and account ID of the calling account.
In the example, the Source ARN designated as `arn:aws:controltower:us-west-2:012345678901:*` is the only ARN allowed to perform the `sts:AssumeRole` action. In other words, only users who can sign in to the account ID `012345678901`, in the `us-west-2` Region, are allowed to perform actions that require this specific role and trust relationship for the AWS Control Tower service, designated as `controltower.amazonaws.com`.

The next example shows the `aws:SourceAccount` and `aws:SourceArn` conditions applied to the role trust policy.

```json
{
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": ["controltower.amazonaws.com"]
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "012345678901"
      },
      "StringLike": {
        "aws:SourceArn": "arn:aws:controltower:us-west-2:012345678901:*"
      }
    }
  }
]
}
```

The example illustrates the `aws:SourceArn` condition statement, with an added `aws:SourceAccount` condition statement. For more information, see Prevent cross-service impersonation (p. 1396).

For general information about permission policies in AWS Control Tower see Manage access to resources (p. 1390).

**Recommendations:**

We recommend that you add conditions to the roles that AWS Control Tower creates, because those roles are directly assumed by other AWS services. For more information, see the example for `AWSControlTowerAdmin`, shown previously in this section. For the AWS Config recorder role, we
recommend adding the `aws:SourceArn` condition, specifying the Config recorder ARN as the permitted source ARN.

For roles such as **AWSControlTowerExecution** or the **roles that can be assumed by the AWS Control Tower Audit account in all managed accounts** (p. 92), we recommend that you add the `aws:PrincipalOrgID` condition to the trust policy for these roles, which validates that the principal accessing the resource belongs to an account in the correct AWS organization. Do not add the `aws:SourceArn` condition statement, because it will not work as expected.

**Note**
In case of drift, it is possible that an AWS Control Tower role may be reset under certain circumstances. It is recommended that you re-check the roles periodically, if you have customized them.

## AWS Control Tower ConfigRecorderRole

AWS Control Tower deploys this role as a resource in the log archive account, the audit account, and in each account created by Account Factory. The role can be assumed by AWS Config, as shown in the role trust relationship artifact, given later in this section. This role is over 1000 lines long, because it allows multiple actions by many AWS services. The role grants permission to AWS Config to record configurations and deliver them to the delivery channels.

**Note**
When you create this IAM role, you give AWS Control Tower permission to manage the AWS Config resources as defined in the permissions policy for this role. The first time that AWS Control Tower uses this role, AWS Config might create a new service-linked role in your account. That role grants AWS Config access to other AWS resources that are required to complete your original AWS Control Tower request.
To learn more about how AWS Config or other services create and use service-linked roles, see [AWS Services That Work with IAM](https://docs.aws.amazon.com/IAM/latest/userguide/servicelinkedrole-select-service.html). Look for the services that have Yes in the Service-Linked Role column to indicate that they support using service-linked roles. Choose a Yes with a link to view the service-linked role documentation for that service.
For a definition of AWS service-linked role, see [AWS service-linked role](https).

**Role name:** aws-controltower-ConfigRecorderRole

**Deployed in these accounts:** Log archive, Audit, Account factory accounts

**Assumed by:** AWS Config

You can view the details and JSON artifacts of the AWS managed policies for this role.

- [ReadOnlyAccess](#)
- To view the complete JSON policy artifact for the AWS Control Tower ConfigRecorderRole, see [AWS_ConfigRole](#).

**Role trust relationship**

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "config.amazonaws.com"
      }
    }
  ]
}
```
How AWS Control Tower aggregates AWS Config rules in unmanaged OUs and accounts

The AWS Control Tower management account creates an organization-level aggregator, which assists in detecting external AWS Config rules, so that AWS Control Tower does not need to gain access to unmanaged accounts. The AWS Control Tower console shows you how many externally created AWS Config rules you have for a given account. You can view details about those external rules in the External Config Rule Compliance tab of the Account details page.

To create the aggregator, AWS Control Tower adds a role with the permissions required to describe an organization and list the accounts under it. The AWSControlTowerConfigAggregatorRoleForOrganizations role requires the AWSConfigRoleForOrganizations managed policy and a trust relationship with config.amazonaws.com.

Here is the IAM policy (JSON artifact) attached to the role:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "organizations:ListAccounts",
        "organizations:DescribeOrganization",
        "organizations:ListAWSServiceAccessForOrganization"
      ],
      "Resource": "*"
    }
  ]
}
```

Here is the AWSControlTowerConfigAggregatorRoleForOrganizations trust relationship:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "config.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

To deploy this functionality in the management account, the following permissions are added in the managed policy AWSControlTowerServiceRolePolicy, which is used by the AWSControlTowerAdmin role when it creates the AWS Config aggregator:
Programmatic roles and trust relationships for the AWS Control Tower audit account

You can sign into the audit account and assume a role to review other accounts programmatically. The audit account does not allow you to log in to other accounts manually.

The audit account gives you programmatic access to other accounts, by means of some roles that are granted to AWS Lambda functions only. For security purposes, these roles have trust relationships with other roles, which means that the conditions under which the roles can be utilized are strictly defined.

The AWS Control Tower stack set StackSet-AWSControlTowerBP-BASELINE-ROLES creates these programmatic-only, cross-account roles in the audit account:

- `aws-controltower-AdministratorExecutionRole`
- `aws-controltower-AuditAdministratorRole`
- `aws-controltower-ReadOnlyExecutionRole`
- `aws-controltower-AuditReadOnlyRole`

`ReadOnlyExecutionRole`: Note that this role allows the audit account to read objects in Amazon S3 buckets across the entire organization (in contrast to the SecurityAudit policy, which allows for metadata access only).
aws-controltower-AdministratorExecutionRole:
- Has administrator permissions
- Cannot be assumed from the console
- Can be assumed only by a role in the audit account – the aws-controltower-AuditAdministratorRole

The following artifact shows the trust relationship for aws-controltower-AdministratorExecutionRole. The placeholder number 012345678901 will be replaced by the Audit_acct_ID number for your audit account.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::012345678901:role/aws-controltower-AuditAdministratorRole"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

aws-controltower-AuditAdministratorRole:
- Can be assumed by the AWS Lambda service only
- Has permission to perform read (Get) and write (Put) operations on Amazon S3 objects with names that start with the string log

Attached policies:
1. AWSLambdaExecute – AWS managed policy
2. AssumeRole-aws-controltower-AuditAdministratorRole – inline policy – Created by AWS Control Tower, artifact follows.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["sts:AssumeRole"],
      "Resource": ["arn:aws:iam::*:role/aws-controltower-AdministratorExecutionRole"],
      "Effect": "Allow"
    }
  ]
}
```

The following artifact shows the trust relationship for aws-controltower-AuditAdministratorRole:

```json
{
  "Version": "2012-10-17",
}
```
Programmatic roles and trust relationships
for the AWS Control Tower audit account

**aws-controltower-ReadOnlyExecutionRole:**

- Cannot be assumed from the console
- Can be assumed only by another role in the audit account – the AuditReadOnlyRole

The following artifact shows the trust relationship for `aws-controltower-ReadOnlyExecutionRole`. The placeholder number 012345678901 will be replaced by the Audit_acct_ID number for your audit account.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::012345678901:role/aws-controltower-AuditReadOnlyRole"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

**aws-controltower-AuditReadOnlyRole:**

- Can be assumed by the AWS Lambda service only
- Has permission to perform read (Get) and write (Put) operations on Amazon S3 objects with names that start with the string log

**Attached policies:**

1. **AWSLambdaExecute** – AWS managed policy
2. **AssumeRole-aws-controltower-AuditReadOnlyRole** – inline policy – Created by AWS Control Tower, artifact follows.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": ["sts:AssumeRole"],
            "Resource": ["arn:aws:iam::*:role/aws-controltower-ReadOnlyExecutionRole"],
            "Effect": "Allow"
        }
    ]
}
```
The following artifact shows the trust relationship for `aws-controltower-AuditAdministratorRole`:

```json
{}
```

Automated Account Provisioning With IAM Roles

To configure Account Factory accounts in a more automated way, you can create Lambda functions in the AWS Control Tower management account, which [assumes the AWSControlTowerExecution role](https://docs.aws.amazon.com/controltower/latest/userguide/iam-access-keys.html) in the member account. Then, using the role, the management account performs the desired configuration steps in each member account.

If you're provisioning accounts using Lambda functions, the identity that will perform this work must have the following IAM permissions policy, in addition to AWSServiceCatalogEndUserFullAccess.

```json
{}
```
The permissions `sso:GetPeregrineStatus`, `sso:ProvisionApplicationInstanceForAWSAccount`, `sso:ProvisionApplicationProfileForAWSAccountInstance`, and `sso:ProvisionSAMLProvider` are required by AWS Control Tower Account Factory to interact with AWS IAM Identity Center.
How AWS Regions Work With AWS Control Tower

Currently, AWS Control Tower is supported in the following AWS Regions:

- US East (N. Virginia)
- US East (Ohio)
- US West (Oregon)
- Canada (Central)
- Asia Pacific (Sydney)
- Asia Pacific (Singapore)
- Europe (Frankfurt)
- Europe (Ireland)
- Europe (London)
- Europe (Stockholm)
- Asia Pacific (Mumbai)
- Asia Pacific (Seoul)
- Asia Pacific (Tokyo)
- Europe (Paris)
- South America (São Paulo)
- US West (N. California)
- Asia Pacific (Hong Kong)
- Asia Pacific (Jakarta)
- Asia Pacific (Osaka)
- Europe (Milan)
- Africa (Cape Town)
- Middle East (Bahrain)
- Israel (Tel Aviv)
- Middle East (UAE)
- Europe (Spain)
- Asia Pacific (Hyderabad)
- Europe (Zurich)
- Asia Pacific (Melbourne)

About your home Region

When you create a landing zone, the Region that you're using for access to the AWS Management console becomes your home AWS Region for AWS Control Tower. During the creation process, some resources are provisioned in the home Region. Other resources, such as OUs and AWS accounts, are global.

After you've selected a home Region, you cannot change it.
Controls and Regions

Currently, all preventive controls work globally. Detective and proactive controls, however, only work in Regions where AWS Control Tower is supported. For more information about the behavior of controls when you activate AWS Control Tower in a new Region, see Configure your AWS Control Tower Regions (p. 98).

Configure your AWS Control Tower Regions

This section describes the behavior you can expect when you extend your AWS Control Tower landing zone into a new AWS Region, or remove a Region from your landing zone configuration. Generally, this action is performed through the Update function of the AWS Control Tower console.

**Note**

We recommend that you avoid expanding your AWS Control Tower landing zone into AWS Regions in which you do not require your workloads to run. Opting out of a Region does not prevent you from deploying resources in that Region, but those resources will remain outside of AWS Control Tower governance.

During configuration of a new Region, AWS Control Tower updates the landing zone, which means that it baselines your landing zone —

- to operate actively in all newly-selected Regions, and
- to cease governing resources in deselected Regions.

Individual accounts within your organizational units (OUs) that are managed by AWS Control Tower are not updated as part of this landing zone update process. Therefore, you must update your accounts by re-registering your OUs.

When configuring your AWS Control Tower Regions, be aware of the following recommendations and limitations:

- Select Regions in which you plan to host AWS resources or workloads.
- Opting out of a Region does not prevent you from deploying resources in that Region, but those resources will remain outside of AWS Control Tower governance.

When you configure your landing zone for new Regions, AWS Control Tower detective controls adhere to the following rules:

- **What exists stays the same.** Guardrail behavior, detective as well as preventive, is unchanged for existing accounts, in existing OUs, in existing Regions.
- **You can’t apply new detective controls to existing OUs containing accounts that are not updated.** When you’ve configured your AWS Control Tower landing zone into a new Region (by updating your landing zone), you must update existing accounts in your existing OUs before you can enable new detective controls on those OUs and accounts.
- **Your existing detective controls begin working in the newly configured Regions as soon as you update the accounts.** When you update your AWS Control Tower landing zone to configure new Regions and then update an account, the detective controls that already are enabled on the OU will begin working on that account in the newly configured Regions.

Configure AWS Control Tower Regions

1. Sign in to the AWS Control Tower console at https://console.aws.amazon.com/controltower
Avoid mixed governance when configuring Regions

It is important to update all accounts in an OU after you extend AWS Control Tower governance to a new AWS Region, and after you remove AWS Control Tower governance from a Region.

Mixed governance is an undesirable situation that can occur if the controls governing an OU are not a complete match to the controls governing each account within an OU. Mixed governance occurs in an OU if accounts are not updated after AWS Control Tower extends governance to a new AWS Region, or removes governance.

In this situation, certain accounts within an OU may have different controls applied in different Regions, when compared to other accounts in the OU, or when compared to the landing zone's overall governance posture.

In an OU with mixed governance, if you provision a new account, that new account receives the same (updated) Region and OU governance posture as the landing zone. However, existing accounts that are not yet updated do not receive the updated Region governance posture.

In general, mixed governance may create contradictory or inaccurate status indicators in the AWS Control Tower console. For example, during mixed governance, opt-in Regions are shown with Not governed status, in registered OUs, for accounts that are not yet updated.

Note
AWS Control Tower does not permit controls to be enabled during a state of mixed governance.

Behavior of controls during mixed governance

• During mixed governance, AWS Control Tower cannot consistently deploy controls that are based on AWS Config rules (that is, detective controls) in Regions that the OU already shows as Governed, because some accounts in the OU have not been updated. You may receive a FAILED_TO_ENABLE error message.

• During mixed governance, if you extend the landing zone's governance to an opt-in Region while any account in the OU has not yet been updated, the EnableControl API operation on the OU fails for

An alternative method of provisioning or updating individual accounts after configuring new Regions is by using the API framework of Service Catalog and the AWS CLI to update the accounts in a batch process. For more information, see Provision and update accounts using automation (p. 48).
detective and proactive controls. You will receive a FAILED_TO_ENABLE error message, because non-updated member accounts within the OU have not yet been opted into those Regions.

- During mixed governance, controls that are part of the Security Hub Service-managed Standard: AWS Control Tower cannot report compliance accurately in Regions where there is a mismatch between the landing zone configuration and the accounts that are not updated.
- Mixed governance does not change the behavior of SCP-based controls (preventive controls), which apply uniformly to every account in an OU, in every governed Region.

**Note**
Mixed governance is not the same as drift, and it is not reported as drift.

**To repair mixed governance**

- Choose Update account for each account in the OU that shows Update available status on the Organizations page in the console.
- Choose Re-Register OU on the Organizations page, which automatically updates all accounts in the OU, for OUs with fewer than 300 accounts.

### Considerations for activating AWS opt-in Regions

Although most AWS Regions are active by default for your AWS account, certain Regions are activated only when you manually select them. This document refers to those Regions as opt-in Regions. In contrast, Regions that are active by default, as soon as your AWS account is created, are referred to as commercial Regions, or simply, Regions.

The term opt-in has a historical basis. Any AWS Regions introduced after March 20, 2019 are considered to be opt-in Regions. Opt-in Regions have higher security requirements than commercial Regions, regarding the sharing of IAM data through accounts that are active in opt-in Regions. All of the data managed through the IAM service is considered identity data, including users, groups, roles, policies, identity providers, their associated data (for example, X.509 signing certificates or context-specific credentials), and other account-level settings, such as password policy and the account alias.

You can activate opt-in Regions automatically during landing zone setup, by selecting them. Your landing zone becomes active in all selected Regions.

If you choose to select an opt-in Region as your AWS Control Tower home Region, activate it first by following the steps in Enabling a Region, when signed in to the AWS Management Console. To bring your own existing Log Archive and Audit accounts from an opt-in Region, manually activate that Region first.

The AWS opt-in Regions include several Regions in which AWS Control Tower is available:

- Asia Pacific (Hong Kong) Region, ap-east-1
- Asia Pacific (Jakarta) Region, ap-southeast-3
- Asia Pacific (Osaka) Region, ap-northeast-3
- Europe (Milan) Region, eu-south-1
- Africa (Cape Town) Region, af-south-1
- Middle East (Bahrain) Region, me-south-1
- Israel (Tel Aviv), il-central-1
- Middle East (UAE) Region, me-central-1
- Europe (Spain) Region, eu-south-2
- Asia Pacific (Hyderabad) Region, ap-south-2
About opt-in Regions

- Europe (Zurich) Region, eu-central-2
- Asia Pacific (Melbourne) Region ap-southeast-4

AWS Control Tower has some controls that work differently in the opt-in Regions than in commercial Regions. For more information, see Control limitations (p. 28). Here are some considerations to keep in mind as you deploy workloads into opt-in Regions.

**Governing or activating?**

Remember that governing a Region is an action that you can select from the AWS Control Tower console, so that controls can be applied in the Region. Activating or deactivating an opt-in Region is a different action that you can choose in the AWS console, which opens the Region to your account, so that you can deploy resources and workloads in the Region.

**Behavioral considerations**

- If you choose to govern opt-in Regions, we recommend that you do not deactivate (opt-out of) any of your governed opt-in Regions, because it can lead to failure of your workloads. AWS Control Tower does not allow deactivation of a governed Region from within the AWS Control Tower console, but be sure that you do not deactivate governed Regions from a source outside of AWS Control Tower, such as the AWS Billing console or AWS SDK.
- When AWS Control Tower extends governance to an opt-in Region, it activates (opts-in) to the Region in all member accounts. When you remove a Region from governance, AWS Control Tower does not deactivate (opt-out of) the Region in the member accounts.
- During Region deselection, AWS Control Tower skips removing resources from an opt-in Region if that Region was deactivated manually for an account from a source outside AWS Control Tower, such as the AWS Billing console or AWS SDK. We recommend that you remove resources from the Regions you've deactivated, or you may receive unexpected billing charges for those resources.
- If your landing zone is decommissioned, AWS Control Tower cleans up resources in all the governed Regions, including the opt-in Regions. However, AWS Control Tower does not deactivate the opt-in Regions. You can deactivate the opt-in Regions as an additional step after decommissioning.
- If your home Region is an opt-in Region, and if you intend to enroll existing accounts as your Log Archive and Audit accounts, you must manually activate the opt-in Region before you can select it as the home Region for your landing zone. See Enabling a Region.
- If AWS Control Tower is set up with an opt-in Region as your home Region, and if you visit the AWS Control Tower service from the AWS console in any other Region, the console does not redirect you automatically to the home Region.
- The underlying API has capacity limits, which may increase latency from a few minutes to many hours, depending on the number of Regions, accounts, and service load. As a best practice, opt-in only to those the AWS Regions where you will run workloads, and opt-in one Region at a time.

**Important limitations for governance and controls**

- If you currently have enabled an AWS Control Tower control that is not supported in an opt-in Region, you will not be able to extend AWS Control Tower governance into that opt-in Region until the control is supported in that Region. For more information see Control limitations (p. 28).
- If you extend AWS Control Tower governance into an opt-in Region in which a specific control is not supported, you will not be able to enable that control in any Region until the control is supported in all the Regions you are governing with AWS Control Tower. For more information see Control limitations (p. 28).
- If all 22 commercial Regions where AWS Control Tower is available are activated, including opt-in Regions, the upper limit on the number of accounts per organizational unit (OU), when extending governance to an OU, is reduced. The limit is 220 instead of 300 accounts. This reduction is due to StackSet limitations. If you require to extend governance to OUs with more than 220 accounts, reduce the number of activated Regions.
Configure the Region deny control

The Region deny control is unique, because it applies to the landing zone as a whole, rather than to any specific OU. To configure the Region deny control, go to the Landing zone settings page and select Modify settings.

- This setting can be changed at a later time.
- When enabled, this control applies to all registered OUs.
- This control cannot be configured for individual OUs.

**Note**
Before you enable the Region deny control, be sure that you do not have existing resources in these Regions, because you will not have access to your resources after you apply the control. While the control is enabled, you will not be able to deploy resources in the denied Regions.

The Region deny control prohibits access to AWS services, based on your AWS Control Tower Region configuration. It denies access to AWS Regions with status Not Governed. The Region deny control also denies access to Regions in which AWS Control Tower is not available. You cannot deny access to your home Region. Certain global AWS services, such as IAM and AWS Organizations, are exempt from the Region deny control. To learn more, see Deny access to AWS based on the requested AWS Region (p. 1344).

When you enable the control, it applies to all registered, top-level OUs in your hierarchy, and it is inherited by OUs lower in the chain. When you remove the control, it is removed on all registered OUs, all non-governed Regions in AWS Control Tower remain in a Not governed status, and you can deploy resources in Regions outside of AWS Control Tower availability.

- Full control name: Deny access to AWS based on the requested AWS Region
- Guardrail description: Disallows access to unlisted operations in global and regional services outside of the specified Regions.
- This is an elective control with preventive guidance.

To view the template for the Region deny control SCP, see Deny access to AWS based on the requested AWS Region (p. 1344) in the AWS Control Tower Guardrail reference. The AWS Control Tower SCP is similar to the SCP for AWS Organizations, but not identical.

You can determine Regional service endpoints on the Regional services page.
Provision and manage accounts in AWS Control Tower

This chapter includes an overview and procedures for provisioning and managing member accounts in your AWS Control Tower landing zone.

It also includes an overview and procedures for enrolling an existing AWS account into AWS Control Tower.

For more information about accounts in AWS Control Tower, see About AWS accounts in AWS Control Tower (p. 105). For information about enrolling multiple accounts into AWS Control Tower, see Register an existing organizational unit with AWS Control Tower (p. 190).

Note
You can perform up to five (5) account-related operations concurrently, including provisioning, updating, and enrolling.

Methods of provisioning

AWS Control Tower provides several methods for creating and updating member accounts. Some methods are primarily console-based, and some methods are primarily automated.

Overview

The standard way to create member accounts is through Account Factory, a console-based product that’s part of the Service Catalog. If your landing zone is not in a state of drift, you can use Create account as a method to add new accounts from the console, as well as Enroll account to enroll existing AWS accounts into AWS Control Tower.

With Account Factory, you can provision basic accounts, by relying on the AWS Control Tower default settings. You also can provision customized accounts that meet requirements for specialized use cases.

Account Factory Customization (AFC) is a way of provisioning customized accounts from the AWS Control Tower console, and it automates the customization and deployment of your accounts. It allows console-based, automated provisioning, after some one-time setup steps, which eliminates the need to write scripts or set up pipelines. For more information, see Customize accounts with Account Factory Customization (AFC) (p. 129).

Console-based methods:

- Through the Account Factory console that is part of Service Catalog, for basic or customized accounts. Review Provision and manage accounts with Account Factory (p. 121) for details and instructions.
- Through the Enroll account feature within AWS Control Tower, if your landing zone is not in a state of drift. See Enroll an existing account (p. 110).
- In the AWS Control Tower console, you can use Account Factory to create, update, or enroll up to five accounts at the same time.

Automated methods:

- Lambda code: From your AWS Control Tower landing zone's management account, using Lambda code and appropriate IAM roles. See Automated Account Provisioning With IAM Roles (p. 95).
What happens when AWS Control Tower creates an account

New accounts in AWS Control Tower are created and then provisioned by an interaction among AWS Control Tower, AWS Organizations, and AWS Service Catalog. For steps to create an account through the AWS Control Tower console, see Enroll an existing account (p. 110).

Behind the scenes of account creation

1. You initiate the request, for example, from the AWS Control Tower Account Factory page, or directly from the Service Catalog console, or by calling the Service Catalog ProvisionProduct API.
2. Service Catalog calls AWS Control Tower.
3. AWS Control Tower begins a workflow, which as a first step calls the AWS Organizations CreateAccount API.
4. After AWS Organizations creates the account, AWS Control Tower completes the provisioning process by applying blueprints and controls.
5. Service Catalog continues to poll AWS Control Tower to check for completion of the provisioning process.
6. When the workflow in AWS Control Tower is complete, Service Catalog finalizes the account’s state and informs you (the requester) of the result.

Permissions required

The permissions required for each method of provisioning and updating are discussed in each section, respectively. With the appropriate user group permissions, provisioners can specify standardized baselines and network configurations for any accounts in their organization.

Note
When provisioning an account, the account requester always must have the CreateAccount and the DescribeCreateAccountStatus permissions. This permission set is part of the Admin role, and it is given automatically when a requester assumes the Admin role. If you delegate permission to provision accounts, you may need to add these permissions directly for the account requestors.

For general information about permissions required in AWS Control Tower, see Using identity-based policies (IAM policies) for AWS Control Tower (p. 1397). For information about roles and accounts in AWS Control Tower, see How AWS Control Tower works with roles to create and manage accounts (p. 87).

Security for your accounts
You can find guidance about best practices to protect the security of your AWS Control Tower management account and member accounts in the AWS Organizations documentation.
About AWS accounts in AWS Control Tower

An AWS account is the container for all your owned resources. These resources include the AWS Identity and Access Management (IAM) identities accepted by the account, which determine who has access to that account. IAM identities can include users, groups, roles, and more. For more information about working with IAM, users, roles, and policies in AWS Control Tower, see Identity and access management in AWS Control Tower.

Resources and account creation time

When AWS Control Tower creates or enrolls an account, it deploys the minimum necessary resource configuration for the account, including resources in the form of Account Factory templates and other resources in your landing zone. These resources may include IAM roles, AWS CloudTrail trails, Service Catalog provisioned products, and IAM Identity Center users. AWS Control Tower also deploys resources, as required by the control configuration, for the organizational unit (OU) in which the new account is destined to become a member account.

AWS Control Tower orchestrates the deployment of these resources on your behalf. It may require several minutes per resource to complete the deployment, so consider the total time before you create or enroll an account. For more information about managing resources in your accounts, see Guidance for creating and modifying AWS Control Tower resources (p. 40).

View your accounts

The Organization page lists all OUs and accounts in your organization, regardless of OU or enrollment status in AWS Control Tower. You can view and enroll member accounts into AWS Control Tower—individually or by OU groups—if each account meets the prerequisites for enrollment.

To view a specific account on the Organization page, you can choose Accounts only from the dropdown menu at the upper right, and then select the name of your account from the table. Alternatively, you can select the name of the parent OU from the table, and you can view a list of all accounts within that OU on the Details page for that OU.

On the Organization page and the Account details page, you can see the account’s State, which is one of these:

- **Not enrolled** – The account is a member of the parent OU, but it is not fully managed by AWS Control Tower. If the parent OU is registered, the account is governed by the preventive controls configured for its registered parent OU, but the OU’s detective controls do not apply to this account. If the parent OU is unregistered, no controls apply to this account.
- **Enrolling** – The account is being brought into governance by AWS Control Tower. We are aligning the account with the control configuration for the parent OU. This process may require several minutes per account resource.
- **Enrolled** – The account is governed by the controls configured for its parent OU. It is fully managed by AWS Control Tower.
- **Enrollment failed** – The account could not be enrolled in AWS Control Tower. For more information, see Common causes for failure of enrollment (p. 111).
- **Update available** – The account has an update available. Accounts in this state are still Enrolled, but the account must be updated to reflect recent changes made to your environment. To update a single account, navigate to the account detail page and select Update account.
If you have multiple accounts with this state under a single OU, you can choose to Re-register the OU and update those accounts together.

About the shared accounts

Three special AWS accounts are associated with AWS Control Tower; the management account, the audit account, and the log archive account. These accounts usually are referred to as shared accounts, or sometimes as core accounts.

- You can select customized names for the audit and log archive accounts when you're setting up your landing zone. For information about changing an account name, see Externally changing AWS Control Tower resource names.
- You also can specify an existing AWS account as an AWS Control Tower security or logging account, during the initial landing zone setup process. This option eliminates the need for AWS Control Tower to create new, shared accounts. (This is a one-time selection.)

For more information about the shared accounts and their associated resources, see What Are the Shared Accounts? (p. 3)

Considerations for bringing existing security or logging accounts

Before accepting an AWS account as a security or logging account, AWS Control Tower checks the account for resources that conflict with AWS Control Tower requirements. For example, you may have a logging bucket with the same name that AWS Control Tower requires. Also, AWS Control Tower validates that the account can provision resources; for example, by ensuring that AWS Security Token Service (AWS STS) is enabled, that the account is not suspended, and that AWS Control Tower has permission to provision resources within the account.

AWS Control Tower does not remove any existing resources in the logging and security accounts that you provide. However, if you choose to enable the AWS Region deny capability, the Region deny control prevents access to resources in denied Regions.

Management account

This AWS account launches AWS Control Tower. By default, the root user for this account and the IAM user or IAM administrator user for this account have full access to all resources within your landing zone.

Note
As a best practice, we recommend signing in as an IAM Identity Center user with Administrator privileges when performing administrative functions within the AWS Control Tower console, instead of the signing in as the root user or IAM administrator user for this account.

For more information about the roles and resources available in the management account, see What is the management account? (p. 4)

Log archive account

The log archive shared account is set up automatically when you create your landing zone.

This account contains a central Amazon S3 bucket for storing a copy of all AWS CloudTrail and AWS Config log files for all other accounts in your landing zone. As a best practice, we recommend restricting log archive account access to teams responsible for compliance and investigations, and their related security or audit tools. This account can be used for automated security audits, or to host custom AWS Config Rules, such as Lambda functions, to perform remediation actions.
For more information about the roles and resources available in the log archive account, see [What is the log archive account? (p. 6)](#)

**Note**
These logs cannot be changed. All logs are stored for the purposes of audit and compliance investigations related to account activity.

## Audit account

This shared account is set up automatically when you create your landing zone.

The audit account should be restricted to security and compliance teams with auditor (read-only) and administrator (full-access) cross-account roles to all accounts in the landing zone. These roles are intended to be used by security and compliance teams to:

- Perform audits through AWS mechanisms, such as hosting custom AWS Config rule Lambda functions.
- Perform automated security operations, such as remediation actions.

The audit account also receives notifications through the Amazon Simple Notification Service (Amazon SNS) service. Three categories of notification can be received:

- **All Configuration Events** – This topic aggregates all CloudTrail and AWS Config notifications from all accounts in your landing zone.
- **Aggregate Security Notifications** – This topic aggregates all security notifications from specific CloudWatch events, AWS Config Rules compliance status change events, and GuardDuty findings.
- **Drift Notifications** – This topic aggregates all the drift warnings discovered across all accounts, users, OUs, and SCPs in your landing zone. For more information on drift, see [Detect and resolve drift in AWS Control Tower](#).

Audit notifications that are triggered within a member account also can send alerts to a local Amazon SNS topic. This functionality allows account administrators to subscribe to audit notifications that are specific to an individual member account. As a result, administrators can resolve issues that affect an individual account, while still aggregating all account notifications to your centralized audit account. For more information, see [Amazon Simple Notification Service Developer Guide](#).

For more information about the roles and resources available in the audit account, see [What is the audit account? (p. 8)](#) Also see [Programmatic roles and trust relationships for the AWS Control Tower audit account](#).

**Important**
The email address you provided for the audit account receives **AWS Notification - Subscription Confirmation** emails from every AWS Region supported by AWS Control Tower. To receive compliance emails in your audit account, you must choose the **Confirm subscription** link within each email from each AWS Region supported by AWS Control Tower.

## About member accounts

Member accounts are the accounts through which your users perform their AWS workloads. These member accounts can be created in Account Factory, by IAM Identity Center users with **Admin** privileges in the Service Catalog console, or by automated methods. When created, these member accounts exist in an OU that was created in the AWS Control Tower console, or registered with AWS Control Tower. For more information, see these related topics:

- [Provision and manage accounts with Account Factory](#)
- [Automate tasks in AWS Control Tower](#)
- [AWS Organizations Terminology and Concepts](#) in the **AWS Organizations User Guide**.
Also see Provision accounts with AWS Control Tower Account Factory for Terraform (AFT) (p. 139).

**Accounts and controls**
Member accounts can be enrolled in AWS Control Tower, or they can be unenrolled. Controls apply differently to enrolled and unenrolled accounts, and controls may apply to accounts in nested OUs based on inheritance.

### Enroll an existing AWS account

You can extend AWS Control Tower governance to an individual, existing AWS account when you enroll it into an organizational unit (OU) that's already governed by AWS Control Tower. Eligible accounts exist in unregistered OUs that are part of the same AWS Organizations organization as the AWS Control Tower OU.

**Note**
You cannot enroll an existing account to serve as your audit or log archive account except during initial landing zone setup.

**Set up trusted access first**

Before you can enroll an existing AWS account into AWS Control Tower you must give permission for AWS Control Tower to manage, or govern, the account. Specifically, AWS Control Tower requires permission to establish trusted access between AWS CloudFormation and AWS Organizations on your behalf, so that AWS CloudFormation can deploy your stack automatically to the accounts in your selected organization. With this trusted access, the AWSControlTowerExecution role conducts activities required to manage each account. That's why you must add this role to each account before you enroll it.

When trusted access is enabled, AWS CloudFormation can create, update, or delete stacks across multiple accounts and AWS Regions with a single operation. AWS Control Tower relies on this trust capability so it can apply roles and permissions to existing accounts before it moves them into a registered organizational unit, and thereby brings them under governance.

To learn more about trusted access and AWS CloudFormation StackSets, see AWS CloudFormationStackSets and AWS Organizations.

### What happens during account enrollment

During the enrollment process, AWS Control Tower performs these actions:

- Baselines the account, which includes deploying these stack sets:
  - AWSControlTowerBP-BASELINE-CLOUDTRAIL
  - AWSControlTowerBP-BASELINE-CLOUDWATCH
  - AWSControlTowerBP-BASELINE-CONFIG
  - AWSControlTowerBP-BASELINE-ROLES
  - AWSControlTowerBP-BASELINE-SERVICE-ROLES
  - AWSControlTowerBP-BASELINE-SERVICE-LINKED-ROLES
  - AWSControlTowerBP-VPC-ACCOUNT-FACTORY-V1

  It is a good idea to review the templates of these stack sets and make sure that they don't conflict with your existing policies.

- Identifies the account through AWS IAM Identity Center or AWS Organizations.

- Places the account into the OU that you've specified. Be sure to apply all SCPs that are applied in the current OU, so that your security posture remains consistent.

- Applies mandatory controls to the account by means of the SCPs that apply to the selected OU as a whole.
Enrolling existing accounts with VPCs

AWS Control Tower handles VPCs differently when you provision a new account in Account Factory than when you enroll an existing account.

- When you create a new account, AWS Control Tower automatically removes the AWS default VPC and creates a new VPC for that account.
- When you enroll an existing account, AWS Control Tower does not create a new VPC for that account.
- When you enroll an existing account, AWS Control Tower does not remove any existing VPC or AWS default VPC associated with the account.

Tip
You can change the default behavior for new accounts by configuring Account Factory, so it does not set up a VPC by default for accounts in your organization under AWS Control Tower. For more information, see Create an Account in AWS Control Tower Without a VPC (p. 1433).

Prerequisites for enrollment

These prerequisites are required before you can enroll an existing AWS account in AWS Control Tower:

1. To enroll an existing AWS account, the AWSControlTowerExecution role must be present in the account you are enrolling. You can review Enroll an account for details and instructions.
2. In addition to the AWSControlTowerExecution role, the existing AWS account you want to enroll must have the following permissions and trust relationships in place. Otherwise, enrollment will fail.

   Role Permission: AdministratorAccess (AWS managed policy)

   Role Trust Relationship:

   ```json
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
```
3. We recommend that the account should not have an AWS Config configuration recorder or delivery channel. These may be deleted or modified through the AWS CLI before you can enroll an account. Otherwise, review Enroll accounts that have existing AWS Config resources for instructions on how you can modify your existing resources.

4. The account that you wish to enroll must exist in the same AWS Organizations organization as the AWS Control Tower management account. The account that exists can be enrolled only into the same organization as the AWS Control Tower management account, in an OU that already is registered with AWS Control Tower.

To check other prerequisites for enrollment, see [Getting Started with AWS Control Tower](#).

**Note**

When you enroll an account into AWS Control Tower, your account is governed by the AWS CloudTrail trail for the AWS Control Tower organization. If you have an existing deployment of a CloudTrail trail, you may see duplicate charges unless you delete the existing trail for the account before you enroll it in AWS Control Tower.

### Enroll an existing account

The **Enroll account** feature is available in the AWS Control Tower console, for enrolling existing AWS accounts so that they are governed by AWS Control Tower. For more information, see [Enroll an existing AWS account](#).

The **Enroll account** capability is available when your landing zone is not in a state of drift. To view this capability in the console:

- Navigate to the **Organization** page in AWS Control Tower.
- Find the name of the account you wish to enroll. To find it, choose **Accounts only** from the dropdown menu at the upper right, and then locate the account name in the filtered table.
- Follow the steps for enrolling an individual account, as shown in the [Steps to enroll an account](#) section.

**Note**

When you are enrolling an existing AWS account, be sure to verify the existing email address. Otherwise, a new account may be created.

Certain errors may require that you refresh the page and try again. If your landing zone is in a state of drift, you may not be able to use the **Enroll account** capability successfully. You'll need to provision new accounts through Account Factory until your landing zone drift has been resolved.

When you enroll accounts from the AWS Control Tower console, you must be signed into an account with a user that has the `AWSServiceCatalogEndUserFullAccess` policy enabled, along with Administrator access permissions to use the AWS Control Tower console, and you cannot be signed in as the root user.

Accounts that you enroll may be updated by means of AWS Service Catalog and the AWS Control Tower account factory, as you would update any other account. Update procedures are given in the section called [Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog](#).
Steps to enroll an account

After the AdministratorAccess permission (policy) is in place in your existing account, follow these steps to enroll the account:

To enroll an individual account in AWS Control Tower

• Navigate to the AWS Control Tower Organization page.
• On the Organization page, accounts that are eligible to be enrolled allow you to select Enroll from the Actions dropdown menu at the top of the section. These accounts also show an Enroll account button when you view them on the Account details page.
• When you choose Enroll account, you'll see an Enroll account page, where you are prompted to add the AWSCloneTowerExecution role to the account. For some instructions, see Manually add the required IAM role to an existing AWS account and enroll it (p. 113).
• Next, select a registered OU from the drop down list. If the account is already in a registered OU, this list will show the OU.
• Choose Enroll account.
• You'll see a modal reminder to add the AWSCloneExecution role and confirm the action.
• Choose Enroll.
• AWS Control Tower begins the process of enrollment, and you are directed back to the Account details page.

Common causes for failure of enrollment

• To enroll an existing account, the AWSCloneExecution role must be present in the account you're enrolling.
• Your IAM principal may lack the necessary permissions to provision an account.
• AWS Security Token Service (AWS STS) is disabled in your AWS account in your home Region, or in any Region supported by AWS Control Tower.
• You may be signed in to an account that needs to be added to the Account Factory Portfolio in AWS Service Catalog. The account must be added before you'll have access to Account Factory so you can create or enroll an account in AWS Control Tower. If the appropriate user or role is not added to the Account Factory portfolio, you'll receive an error when you attempt to add an account. For instructions on how to grant access to AWS Service Catalog portfolios, see Granting access to users.
• You may be signed in as root.
• The account you're trying to enroll may have AWS Config settings that are residual. In particular, the account may have a configuration recorder or delivery channel. These must be deleted or modified through the AWS CLI before you can enroll an account. For more information, see Enroll accounts that have existing AWS Config resources (p. 116) and Interacting with AWS Control Tower using AWS CloudShell (p. 52).
• If the account belongs to another OU with a management account, including another AWS Control Tower OU, you must terminate the account in its current OU before it can join another OU. Existing resources must be removed in the original OU. Otherwise, enrollment will fail.
• Account provisioning and enrollment fails if your destination OU's SCPs don't allow you to create all of the resources required for that account. For example, an SCP in your destination OU may block resource creation without certain tags. In this case, account provisioning or enrollment fails, because AWS Control Tower does not support tagging of resources. For help, contact your account representative, or AWS Support.

For more information about how AWS Control Tower works with roles when you're creating new accounts or enrolling existing accounts, see How AWS Control Tower works with roles to create and manage accounts (p. 87).
What if the account does not meet the prerequisites?

Tip
If you cannot confirm that an existing AWS account meets the enrollment prerequisites, you can set up an Enrollment OU and enroll the account into that OU. After enrollment is successful, you can move the account to the desired OU. If enrollment happens to fail, no other accounts or OUs are affected by the failure.

If you have doubts that your existing accounts and their configurations are compatible with AWS Control Tower, you can follow the best practice recommended in the following section.

Recommended: You can set up a two-step approach to account enrollment

- First, use an AWS Config conformance pack to evaluate how your accounts may be affected by some AWS Control Tower controls. To determine how enrollment into AWS Control Tower may affect your accounts, see Extend AWS Control Tower governance using AWS Config conformance packs.
- Next, you may wish to enroll the account. If the compliance results are satisfactory, the migration path is easier because you can enroll the account without unexpected consequences.
- After you've done your evaluation, if you decide to set up an AWS Control Tower landing zone, you may need to remove the AWS Config delivery channel and configuration recorder that were created for your evaluation. Then you'll be able to set up AWS Control Tower successfully.

Note
The conformance pack also works in situations where the accounts are located in OUs registered by AWS Control Tower, but the workloads run within AWS Regions that don't have AWS Control Tower support. You can use the conformance pack to manage resources in accounts that exist in Regions where AWS Control Tower is not deployed.

What if the account does not meet the prerequisites?

Remember that, as a prerequisite, accounts eligible to be enrolled into AWS Control Tower governance must be part of the same overall organization. To fulfill this prerequisite for account enrollment, you can follow these preparatory steps to move an account into the same organization as AWS Control Tower.

Preparatory steps to bring an account into the same organization as AWS Control Tower

1. Drop the account from its existing organization. You must provide a separate payment method if you use this approach.
2. Invite the account to join the AWS Control Tower organization. For more information, see Inviting an AWS account to join your organization in the AWS Organizations User Guide.
3. Accept the invitation. The account shows up in the root of the organization. This step moves the account into the same organization as AWS Control Tower and establishes SCPs and consolidated billing.

Tip
You can send the invitation for the new organization before the account drops out of the old organization. The invitation will be waiting when the account officially drops out of its existing organization.

Steps to fulfill the remaining prerequisites:

1. Create the necessary AWSControlTowerExecution role.
2. Clear out the default VPC. (This part is optional. AWS Control Tower doesn't change your existing default VPC.)
3. Delete or modify any existing AWS Config configuration recorder or delivery channel through the AWS CLI or AWS CloudShell. For more information, see Example AWS Config CLI commands for resource status (p. 113) and Enroll accounts that have existing AWS Config resources (p. 116)
After you've completed these preparatory steps, you can enroll the account into AWS Control Tower. For more information, see Steps to enroll an account (p. 111). This step brings the account into full AWS Control Tower governance.

Optional steps to deprovision an account, so it can be enrolled and keep its stack

1. To keep the applied AWS CloudFormation stack, delete the stack instance from the stack sets, and choose Retain stacks for the instance.
2. Terminate the account provisioned product in AWS Service Catalog Account Factory. (This step only removes the provisioned product from AWS Control Tower. It doesn't delete the account.)
3. Set up the account with the necessary billing details, as required for any account that doesn't belong to an organization. Then remove the account from the organization. (You do this, so the account doesn't count against the total in your AWS Organizations quota.)
4. Clean up the account if resources remain, and then close it, following the account closure steps in Unmanage an account (p. 126).
5. If you have a Suspended OU with defined controls, you can move the account there instead of doing Step 1.

Example AWS Config CLI commands for resource status

Here are some example AWS Config CLI commands you can use to determine the status of your configuration recorder and delivery channel.

View commands:

- `aws configservice describe-delivery-channels`
- `aws configservice describe-delivery-channel-status`
- `aws configservice describe-configuration-recorders`

The normal response is something like "name": "default"

Delete commands:

- `aws configservice stop-configuration-recorder --configuration-recorder-name NAME-FROM-DESCRIBE-OUTPUT`
- `aws configservice delete-delivery-channel --delivery-channel-name NAME-FROM-DESCRIBE-OUTPUT`
- `aws configservice delete-configuration-recorder --configuration-recorder-name NAME-FROM-DESCRIBE-OUTPUT`

Manually add the required IAM role to an existing AWS account and enroll it

If you’ve already set up your AWS Control Tower landing zone, you can begin enrolling your organization’s accounts into an OU that is registered with AWS Control Tower. If you haven't set up your landing zone, follow the steps as described in the AWS Control Tower User Guide at Getting Started, Step 2. After the landing zone is ready, complete the following steps to bring existing accounts into governance by AWS Control Tower, manually.

Be sure to review the Prerequisites for enrollment (p. 109) noted previously in this chapter.
Before enrolling an account with AWS Control Tower, you must give AWS Control Tower permission to manage that account. To do so, you'll add a role that has full access to the account, as shown in the steps that follow. These steps must be performed for each account that you enroll.

**For each account:**

**Step 1: Sign in with administrator access to the management account of the organization that currently contains the account you wish to enroll.**

For example, if you created this account from AWS Organizations and you use a cross-account IAM role to sign in, then you may follow these steps:

1. Sign in to your organization's management account.
2. Go to IAM.
3. Go to Roles.
4. Choose Create role.
5. When asked to select which service the role is for, select EC2 and choose Next:Permissions. You will change this to AWS Control Tower later.
6. When asked to attach policies, choose AdministratorAccess.
7. Choose Next:Tags.
8. You may see an optional screen titled Add tags. Skip this screen for now by choosing Next:Review.
10. Enter a brief description in the Description box, such as Allows full account access for enrollment.
11. Choose Create role.
12. Navigate to the role you just created. Choose Roles on the left. Select AWSControlTowerExecution.
14. Copy the code example shown here and paste it into the Policy Document. Replace the string Management Account ID with the actual management account ID of your management account. Here is the policy to paste:

```json
{
  "Version":"2012-10-17",
  "Statement": [ 
    { 
      "Effect":"Allow",
      "Action": [ "sts:AssumeRole" ],
      "Condition": { 
        "StringEquals": { 
          "sts:PrincipalArn": Management Account ID
        }
      }
    }
  ]
}
```
Step 3: Enroll the account by moving it into a registered OU, and verify enrollment.

After you’ve set up the necessary permissions by creating the role, follow these steps to enroll the account and verify enrollment.

1. **Sign in again as Admin and go to AWS Control Tower.**
2. **Enroll the account.**
   - From the **Organization** page in AWS Control Tower, select your account, then choose **Enroll** from the **Actions** dropdown menu at the upper right.
   - Follow the steps for enrolling an individual account, as shown on the **Steps to enroll an account** (p. 111) page.
3. **Verify enrollment.**
   - From AWS Control Tower, choose **Organization** in the left navigation.
   - Look for the account you have recently enrolled. Its initial state will show a status of **Enrolling**.
   - When the state changes to **Enrolled**, the move was successful.

To continue this process, sign into each account in your organization that you want to enroll in AWS Control Tower. Repeat the prerequisite steps and the enrollment steps for each account.

### Automated enrollment of AWS Organizations accounts

You can use the enrollment method described in a blog post called **Enroll existing AWS accounts into AWS Control Tower** to enroll your AWS Organizations accounts into AWS Control Tower with a programmatic process.

The following YAML template may assist you in creating the required role in an account, so that it can be enrolled programatically.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure the AWSControlTowerExecution role to enable use of your account as a target account in AWS CloudFormation StackSets.
Parameters:
  AdministratorAccountId:
    Type: String
    Description: AWS Account Id of the administrator account (the account in which StackSets will be created).
    MaxLength: 12
    MinLength: 12
Resources:
  ExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: AWSControlTowerExecution
      AssumeRolePolicyDocument:
```
Enroll accounts that have existing AWS Config resources

This topic provides a step-by-step approach for how to enroll accounts that have existing AWS Config resources. For examples of how to check your existing resources, see Example AWS Config CLI commands for resource status (p. 113).

Note
If you plan to bring existing AWS accounts into AWS Control Tower as Audit and Log archive accounts, and if those accounts have existing AWS Config resources, you must delete or modify the existing AWS Config resources before you can enroll the accounts into AWS Control Tower.

Examples of AWS Config resources

Here are some types of AWS Config resources that your account could have already. These resources may need to be modified so that you can enroll your account into AWS Control Tower.

- AWS Config recorder
- AWS Config delivery channel
- AWS Config aggregation authorization

Assumptions

- You have deployed an AWS Control Tower landing zone
- Your account is not enrolled with AWS Control Tower already.
- Your account has at least one pre-existing AWS Config resource in at least one of the AWS Control Tower Regions governed by the management account.
- Your account is not the AWS Control Tower management account.
- Your account is not in governance drift.

For a blog that describes an automated approach to enrolling accounts with existing AWS Config resources, see Automate enrollment of accounts with existing AWS Config resources into AWS Control Tower. You'll be able to submit a single support ticket for all of the accounts you wish to enroll, as described in Step 1: Contact customer support with a ticket, to add the account to the AWS Control Tower allow list (p. 117), which follows.

Limitations

- The account can be enrolled only by using the AWS Control Tower workflow for extending governance.
- If the resources are modified and create drift on the account, AWS Control Tower does not update the resources.
AWS Config resources in Regions that are not governed by AWS Control Tower are not changed.

Note
If you attempt to enroll an account that has existing Config resources, without having the account added to the allow list, enrollment will fail. Thereafter, if you subsequently try to add that same account to the allow list, AWS Control Tower cannot validate that the account is provisioned correctly. You must deprovision the account from AWS Control Tower before you can request the allow list and then enroll it. If you only move the account to a different AWS Control Tower OU, it causes governance drift, which also prevents the account from being added to the allow list.

This process has 5 main steps.

1. Add the account to the AWS Control Tower allow list.
2. Create a new IAM role in the account.
3. Modify pre-existing AWS Config resources.
4. Create AWS Config resources in AWS Regions where they don't exist.
5. Enroll the account with AWS Control Tower.

Before you proceed, consider the following expectations regarding this process.

- AWS Control Tower does not create any AWS Config resources in this account.
- After enrollment, AWS Control Tower controls automatically protect the AWS Config resources you created, including the new IAM role.
- If any changes are made to the AWS Config resources after enrollment, those resources must be updated to align with AWS Control Tower settings before you can re-enroll the account.

Step 1: Contact customer support with a ticket, to add the account to the AWS Control Tower allow list

Include this phrase in your ticket subject line:

Enroll accounts that have existing AWS Config resources into AWS Control Tower

Include the following details in the body of your ticket:

- Management account number
- Account numbers of member accounts that have existing AWS Config resources
- Your selected home Region for AWS Control Tower setup

Note
The required time for adding your account to the allow list is 2 business days.

Step 2: Create a new IAM role in the member account

1. Open the AWS CloudFormation console for the member account.
2. Create a new stack using the following template

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config
```
Step 3: Identify the AWS Regions with pre-existing resources

For each governed Region (AWS Control Tower governed) in the account, identify and note the Regions that have at least one of the existing AWS Config resource example types shown previously.

Step 4: Identify the AWS Regions without any AWS Config resources

For each governed Region (AWS Control Tower governed) in the account, identify and note the Regions in which there are no AWS Config resources of the example types shown previously.

Step 5: Modify the existing resources in each AWS Region

For this step, the following information is needed about your AWS Control Tower setup.

- LOGGING_ACCOUNT - the Logging account ID
- AUDIT_ACCOUNT - the Audit account ID
- IAM_ROLE_ARN - the IAM role ARN created in Step 1
- ORGANIZATION_ID - the organization ID for the management account
- MEMBER_ACCOUNT_NUMBER - the member account that is being modified
- HOME_REGION - the home Region for AWS Control Tower setup.

Resources:

CustomerCreatedConfigRecorderRole:
  Type: AWS::IAM::Role
  Properties:
    RoleName: aws-controltower-ConfigRecorderRole-customer-created
    AssumeRolePolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - config.amazonaws.com
          Action:
            - sts:AssumeRole
          Path: /
          ManagedPolicyArns:
            - arn:aws:iam::aws:policy/service-role/AWS_ConfigRole
            - arn:aws:iam::aws:policy/ReadOnlyAccess

3. Provide the name for the stack as CustomerCreatedConfigRecorderRoleForControlTower
4. Create the stack.

Note

Any SCPs that you create should exclude an aws-controltower-ConfigRecorderRole* role. Do not modify the permissions that restrict the ability for AWS Config rules to perform evaluations.

Follow these guidelines so that you don't receive an AccessDeniedException when you have SCPs that block aws-controltower-ConfigRecorderRole* from calling Config.
Step 5a. AWS Config recorder resources

Only one AWS Config recorder can exist per AWS Region. If one exists, modify the settings as shown. Replace the item `GLOBAL_RESOURCE_RECORDING` with `true` in your home Region. Replace the item with `false` for other Regions where an AWS Config recorder exists.

- **Name**: DON'T CHANGE
- **RoleARN**: `IAM_ROLE_ARN`
  - **RecordingGroup**:
    - **AllSupported**: `true`
    - **IncludeGlobalResourceTypes**: `GLOBAL_RESOURCE_RECORDING`
    - **ResourceTypes**: Empty

This modification can be made through the AWS CLI using the following command. Replace the string `RECORDER_NAME` with the existing AWS Config recorder name.

```
aws configservice put-configuration-recorder --configuration-recorder name=RECORDER_NAME,roleARN=arn:aws:iam::MEMBER_ACCOUNT_NUMBER:role/aws-controltower-ConfigRecorderRole-customer-created --recording-group allSupported=true,includeGlobalResourceTypes=GLOBAL_RESOURCE_RECORDING --region CURRENT_REGION
```

Step 5b. Modify AWS Config delivery channel resources

Only one AWS Config delivery channel can exist per Region. If another exists, modify the settings as shown.

- **Name**: DON'T CHANGE
- **ConfigSnapshotDeliveryProperties**: TwentyFour_Hours
- **S3BucketName**: The logging bucket name from the AWS Control Tower logging account
  
  `aws-controltower-logs-LOGGING_ACCOUNT_ID-HOME_REGION`
- **S3KeyPrefix**: `ORGANIZATION_ID`
- **SnsTopicARN**: The SNS topic ARN from the audit account, with the following format:


This modification can be made through the AWS CLI using the following command. Replace the string `DELIVERY_CHANNEL_NAME` with the existing AWS Config recorder name.

```
aws configservice put-delivery-channel --delivery-channel name=DELIVERY_CHANNEL_NAME,s3BucketName=aws-controltower-logs-LOGGING_ACCOUNT_ID-HOME_REGION,s3KeyPrefix="ORGANIZATION_ID",configSnapshotDeliveryProperties={deliveryFrequency=TwentyFour_Hours},snsTopicARN=arn:aws:sns:CURRENT_REGION:AUDIT_ACCOUNT:aws-controltower-AllConfigNotifications --region CURRENT_REGION
```
Step 5c. Modify AWS Config aggregation authorization resources

Multiple aggregation authorizations can exist per Region. AWS Control Tower requires an aggregation authorization that specifies the audit account as the authorized account, and has the home Region for AWS Control Tower as the authorized Region. If it doesn’t exist, create a new one with the following settings:

- **AuthorizedAccountId**: The Audit account ID
- **AuthorizedAwsRegion**: The home Region for the AWS Control Tower setup

This modification can be made through the AWS CLI using the following command:

```bash
aws configservice put-aggregation-authorization --authorized-account-id AUDIT_ACCOUNT_ID --authorized-aws-region HOME_REGION --region CURRENT_REGION
```

Step 6: Create resources where they don’t exist, in Regions governed by AWS Control Tower

Revise the AWS CloudFormation template, so that in your home Region the `IncludeGlobalResourcesTypes` parameter has the value `GLOBAL_RESOURCE_RECORDING`, as shown in the example that follows. Also update the required fields in the template, as specified in this section.

Replace the item `GLOBAL_RESOURCE_RECORDING` with `true` in your home Region. Replace the item with `false` for other Regions where an AWS Config recorder exists.

1. Navigate to the management account’s AWS CloudFormation console.
2. Create a new StackSet with the name `CustomerCreatedConfigResourcesForControlTower`.
3. Copy and update the following template:

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config
Resources:
  CustomerCreatedConfigRecorder:
    Type: AWS::Config::ConfigurationRecorder
    Properties:
      Name: aws-controltower-BaselineConfigRecorder-customer-created
      RoleARN: !Sub arn:aws:iam::${AWS::AccountId}:role/aws-controltower-ConfigRecorderRole-customer-created
      RecordingGroup:
        AllSupported: true
        IncludeGlobalResourceTypes: GLOBAL_RESOURCE_RECORDING
        ResourceTypes: []
  CustomerCreatedConfigDeliveryChannel:
    Type: AWS::Config::DeliveryChannel
    Properties:
      Name: aws-controltower-BaselineConfigDeliveryChannel-customer-created
      ConfigSnapshotDeliveryProperties:
        DeliveryFrequency: TwentyFour_Hours
        S3BucketName: aws-controltower-logs-LOGGING_ACCOUNT-HOME_REGION
        S3KeyPrefix: ORGANIZATION_ID
  CustomerCreatedAggregationAuthorization:
```

---

120
Type: "AWS::Config::AggregationAuthorization"
Properties:
  AuthorizedAccountId: AUDIT_ACCOUNT
  AuthorizedAwsRegion: HOME_REGION

Update the template with required fields:

a. In the S3BucketName field, replace the LOGGING_ACCOUNT_ID and HOME_REGION
b. In the S3KeyPrefix field, replace the ORGANIZATION_ID
c. In the SnsTopicARN field, replace the AUDIT_ACCOUNT
d. In the AuthorizedAccountld field, replace the AUDIT_ACCOUNT
e. In the AuthorizedAwsRegion field, replace the HOME_REGION

4. During deployment on the AWS CloudFormation console, add the member account number.
5. Add the AWS Regions that were identified in Step 4.
6. Deploy the stack set.

**Step 7: Register the OU with AWS Control Tower**

In the AWS Control Tower dashboard, register the OU.

*Note*

The Enroll account workflow will not succeed for this task. You must choose Register OU or Re-register OU.

**Provision and manage accounts with Account Factory**

This chapter includes an overview and procedures for provisioning new member accounts in an AWS Control Tower landing zone with Account Factory.

**Permissions for configuring and provisioning accounts**

The AWS Control Tower Account Factory enables cloud administrators and users in AWS IAM Identity Center to provision accounts in your landing zone. By default, IAM Identity Center users that provision accounts must be in the AWSAccountFactory group or the management group.

*Note*

Exercise caution when working from the management account, as you would when using any account that has permissions across your organization.

The AWS Control Tower management account has a trust relationship with the AWSControlTowerExecution role, which allows account setup from the management account, including some automated account setup. For more information about the AWSControlTowerExecution role, see How AWS Control Tower works with roles to create and manage accounts (p. 87).

*Note*

To enroll an existing AWS account into AWS Control Tower, that account must have the AWSControlTowerExecution role enabled. For more information about how to enroll an existing account, see Enroll an existing AWS account (p. 108).
Provision accounts with AWS Service Catalog Account Factory

The following procedure describes how to create and provision accounts as a user in IAM Identity Center through AWS Service Catalog. This procedure also is referred to as advanced account provisioning, or manual account provisioning. Optionally, you may be able to provision accounts programmatically, with the AWS CLI or with AWS Control Tower Account Factory for Terraform (AFT). You may be able to provision customized accounts in the console if you've previously set up custom blueprints. For more information about customization, see Customize accounts with Account Factory Customization (AFC) (p. 129).

To provision accounts individually in Account Factory, as a user

1. Sign in from your user portal URL.
2. From Your applications, choose AWS Account.
3. From the list of accounts, choose the account ID for your management account. This ID may also have a label, for example, (Management).
4. From AWSServiceCatalogEndUserAccess, choose Management console. This opens the AWS Management Console for this user in this account.
5. Ensure that you've selected the correct AWS Region for provisioning accounts, which should be your AWS Control Tower Region.
6. Search for and choose Service Catalog to open the Service Catalog console.
7. In the navigation pane, choose Products.
8. Select AWS Control Tower Account Factory, then choose the Launch product button. This selection starts the wizard to provision a new account.
9. Fill in the information, and keep the following in mind:
   - The SSOUUserEmail can be a new email address, or the email address associated with an existing IAM Identity Center user. Whichever you choose, this user will have administrative access to the account you’re provisioning.
   - The AccountEmail must be an email address that isn't already associated with an AWS account. If you used a new email address in SSOUUserEmail, you can use that email address here.
10. Don't define TagOptions and don't enable Notifications, otherwise the account can fail to be provisioned. When you're finished, choose Launch product.
11. Review your account settings, and then choose Launch. Don't create a resource plan, otherwise the account will fail to be provisioned.
12. Your account is now being provisioned. It can take a few minutes to complete. You can refresh the page to update the displayed status information.

   Note
   Up to five accounts can be provisioned at a time.

Considerations for managing accounts in Account Factory

You can update, unmanage, and close accounts that you create and provision through Account Factory. You can recycle accounts by updating the user parameters in the accounts that you want to repurpose. You can also change an account's organizational unit (OU).

   Note
   When updating a provisioned product that's associated with an account that Account Factory vends, if you specify a new user email address for AWS IAM Identity Center, AWS Control Tower...
creates a new user in IAM Identity Center. The previously created account isn't removed. For information about removing the previous IAM Identity Center user email address from IAM Identity Center, see Disabling a User.

**Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog**

The easiest way to update an enrolled account is through the AWS Control Tower console. Individual account updates are useful for resolving drift, such as Moved Member Account (p. 173). Account updates also are required as part of a full landing zone update.

If you move an account from one organizational unit (OU) to another, remember that the controls applied by the new OU may be different than the controls in the former OU. Be sure that the controls in the new OU meet your policy requirements for the account.

**Control behavior when accounts are moved between OUs**

When you move an account between OUs, the controls for the destination OU are applied to the account. However, the controls that applied to the account from the former OU are not removed. The exact behavior of the controls is specific to the implementation of the controls that are active on the former OU and the destination OU.

- **For controls implemented with AWS Config rules:** The controls from the previous OU are not removed. These controls must be removed manually.
- **For controls implemented with SCPs:** The controls from the previous OU are not removed. These controls must be removed manually.
- **For controls implemented with AWS CloudFormation hooks:** This behavior depends on the status of controls in the new OU.
  - **If the destination OU has no hook-based controls active:** The old controls remain active for the moved account, unless you remove them manually.
  - **If the destination OU has hook controls active:** The old controls are removed and the controls in the destination OU are applied to the account.

**Update the account in the console**

**To update an account in the AWS Control Tower console**

1. When signed in to AWS Control Tower, navigate to the Organization page.
2. In the list of OUs and accounts, select the name of the account you wish to update. Accounts that are available for updating show a status of Update available.
3. Next you'll see the Account details page for your selected account.
4. In the upper right, choose Update account.

**Update the provisioned product**

The following procedure guides you through how to update your account in Account Factory or move it to a new OU, by updating the account's provisioned product in Service Catalog.
To update an Account Factory account or change its OU through Service Catalog

1. Sign in to the AWS Management Console, and open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/.

   Note
   You must sign in as a user with permissions to provision new products in Service Catalog (for example, an IAM Identity Center user in AWSAccountFactory or AWSServiceCatalogAdmins groups).

2. In the navigation pane, choose Provisioning, and then choose Provisioned products.

3. For each of the member accounts listed, perform the following steps to update all member accounts:

   a. Select a member account. You're directed to the Provisioned product details page for that account.

   b. On the Provisioned product details page, choose the Events tab.

   c. Make a note of the following parameters:

      • SSOUserEmail (Available in provisioned product details)
      • AccountEmail (Available in provisioned product details)
      • SSOUserFirstName (Available in IAM Identity Center)
      • SSOUserLastName (Available in IAM Identity Center)
      • AccountName (Available in IAM Identity Center)

   d. From Actions, choose Update.

   e. Choose the button next to the Version of the product you want to update, and choose Next.

   f. Provide the parameter values that were mentioned previously.

      • If you want to keep the existing OU, for ManagedOrganizationalUnit, choose the OU that the account was already in.
      • If you want to migrate the account to a new OU, for ManagedOrganizationalUnit, choose the new OU for the account.

   A central cloud administrator can find this information in the AWS Control Tower console, on the Organization page.

   g. Choose Next.

   h. Review your changes, and then choose Update. This process can take a few minutes per account.

Change email address of an enrolled account

To change the email address of an enrolled member account in AWS Control Tower, follow the procedure in this section.

   Note
   The following procedure doesn't allow you to change the email address of a management account, log archive account, or audit account. For more information about that, see How do I change the email address associated with my AWS account? or contact AWS Support.

To change the email address of an account that AWS Control Tower creates

1. Recover the root user password for the account. You can follow the steps in the article How do I recover a lost or forgotten AWS password?

2. Sign in to the account with the root user password.
3. Change the email address as you would for any other AWS account, and wait for the change to reflect in AWS Organizations. You might experience a delay while the email address change finishes updating.

4. Update the provisioned product in Service Catalog using the email address that previously belonged to the account. The process for updating the provisioned product includes associating the new email address with the provisioned product. This way the email address change takes effect in AWS Control Tower. Use the new email address for updates to subsequently provisioned products.

To change the password or email address of a member account that you created with AWS Organizations, see Accessing a member account as the root user in the AWS Organizations User Guide.

**Change the name of an enrolled account**

Follow the procedure given in this section to change the name of an enrolled AWS Control Tower account.

**To change the name of an account created by AWS Control Tower**

1. Recover the root password for the account. You can follow the steps outlined in this article, How do I recover a lost or forgotten AWS password?
2. Sign in to the account with the root password.
4. Change the name in Account settings, as you would for any other AWS account.
5. AWS Control Tower automatically updates itself to reflect the name change. This update will not be reflected in the provisioned product in Service Catalog.

**Configure Account Factory with Amazon Virtual Private Cloud settings**

Account Factory allows you to create pre-approved baselines and configuration options for accounts in your organization. You can configure and provision new accounts through AWS Service Catalog.

On the Account Factory page, you can see a list of organizational units (OUs) and their allow list status. By default, all OUs are on the allow list, which means that accounts can be provisioned under them. You can disable certain OUs for account provisioning through AWS Service Catalog.

You can view the Amazon VPC configuration options available to your end users when they provision new accounts.

**To configure Amazon VPC settings in Account Factory**

1. As a central cloud administrator, sign into the AWS Control Tower console with administrator permissions in the management account.
2. From the left side of the dashboard, select Account Factory to navigate to the Account Factory network configuration page. There you can see the default network settings displayed. To edit, select Edit and view the editable version of your Account Factory network configuration settings.
3. You can modify each field of the default settings as needed. Choose the VPC configuration options you’d like to establish for all new Account Factory accounts that your end users may create, and enter your settings into the fields.

- Choose disabled or enabled to create a public subnet in Amazon VPC. By default, the internet-accessible subnet is disallowed.
Unmanage an account

If you created an account in Account Factory or enrolled an AWS account, and you no longer want the account to be managed by AWS Control Tower in a landing zone, you can unmanage the account from the AWS Control Tower console.

When you unmanage an AWS Control Tower account, all resources provisioned by AWS Control Tower are removed, including any blueprints. The account is moved out of any AWS Control Tower OU and into the Root area. The account is no longer part of a registered OU, and it is no longer subject to AWS Control Tower SCPs. You can close the account through AWS Organizations.

Unmanaging an account also can be done in the Service Catalog console by an IAM Identity Center user in the AWSAccountFactory group, by terminating the Provisioned Product. For more information on IAM Identity Center users or groups, see Manage users and access through AWS IAM Identity Center. The following procedure describes how to unmanage a member account in Service Catalog.

To unmanage an enrolled account

2. In the left navigation pane, choose Provisioned products list.
3. From the list of provisioned accounts, choose the name of the account that you want AWS Control Tower no longer to manage.
4. On the Provisioned product details page, from the Actions menu, choose Terminate.
5. From the dialog box that appears, choose Terminate.
6. When the account has been unmanaged, its status changes to Not Enrolled.
7. If you no longer need the account, close it. For more information about closing AWS accounts, see [Closing an account](#) in the [AWS Billing User Guide](#)

When you unmanage a customized account, AWS Control Tower removes the resources that the blueprint has deployed, as well as any other resources that AWS Control Tower created within the account. After you unmanage the account, you can close the account through AWS Organizations.

**Note**

An unmanaged account is not closed or deleted. When the account has been unmanaged, the IAM Identity Center user that you selected when you created the account in Account Factory still has administrative access to the account. If you do not want this user to have administrative access, you must change this setting in IAM Identity Center by updating the account in Account Factory and changing the IAM Identity Center user email address for the account. For more information, see [Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog](#).

**Video Walkthrough**

This video (3:25) describes how to remove an account from AWS Control Tower, gain root access to the account, and finally close the AWS account. You also can close an account with an [AWS Organizations API](#). For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

[Video Walkthrough of Closing an Account in AWS Control Tower](#).

You can view a list of AWS [YouTube videos](#) that explain common tasks in AWS Control Tower.

**Close an account created in Account Factory**

Accounts created in Account Factory are AWS accounts. For information about closing AWS accounts, see [Closing an account](#) in the [AWS Billing User Guide](#).

**Note**

Closing an AWS account is not the same as unmanaging an account from AWS Control Tower—these are separate actions. You must unmanage the account before you close it.

**Close an AWS Control Tower member account through AWS Organizations**

You can close your AWS Control Tower member accounts from your organization's management account without a requirement to sign in to each member account individually with root credentials, by means of AWS Organizations. You cannot close your management account in this way, however.

When you call the AWS Organizations [CloseAccount API](#), or close an account in the AWS Organizations console, the member account is isolated for 90 days, as any AWS account would be. The account shows a Suspended status in AWS Control Tower and AWS Organizations. If you attempt to work with the account during that 90 days, AWS Control Tower gives an error message.

Before the 90 days expire, you can restore the member account, as you can do with any AWS account. After that 90-day time, the account's records are removed.

We recommend, as a best practice, to unmanage a member account before you close that account. If you close a member account without first unmanaging it, AWS Control Tower shows the account's status as Suspended, but also as Enrolled. As a result, if you attempt to Re-register the account's OU during that 90-day time, AWS Control Tower produces an error message. The suspended account essentially blocks
the re-registering actions with a pre-check failure. If you remove the account from the OU, you can **re-register** the OU, but AWS may produce an error regarding a missing method of payment for the account. To work around this constraint, create another OU, and move the account to that OU before you try to re-register. We recommend naming this OU the **Suspended** OU.

**Note**
If you do not unmanage the account before you close it, you must delete the account’s provisioned product in AWS Service Catalog after those 90 days are finished.

For more information, see the AWS Organizations documentation about the [CloseAccount API](https://docs.aws.amazon.com/organizations/latest/userguide/orgs_api.html#OrgsAPI.CloseAccount).

## Resource Considerations for Account Factory

When an account is provisioned with Account Factory, the following AWS resources are created within the account.

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS CloudFormation</td>
<td>Stacks</td>
<td>StackSet-AWSControlTowerBP-BASELINE-CLOUDTRAIL-*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-CONFIG-*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-ROLES-*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerBP-BASELINE-SERVICE-ROLES-*</td>
</tr>
<tr>
<td>AWS CloudTrail</td>
<td>Trail</td>
<td>aws-controltower-BaselineCloudTrail</td>
</tr>
<tr>
<td>Amazon CloudWatch</td>
<td>CloudWatch Event Rules</td>
<td>aws-controltower-ConfigComplianceChangeEventRule</td>
</tr>
<tr>
<td>Amazon CloudWatch</td>
<td>CloudWatch Logs</td>
<td>aws-controltower/CloudTrailLogs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/aws/lambda/aws-controltower-NotificationForwarder</td>
</tr>
<tr>
<td>AWS Identity and Access</td>
<td>Roles</td>
<td>aws-controltower-AdministratorExecutionRole</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td>aws-controltower-CloudWatchLogsRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-ConfigRecorderRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-ForwardSnsNotificationRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-controltower-ReadOnlyExecutionRole</td>
</tr>
</tbody>
</table>
Customize accounts with Account Factory Customization (AFC)

AWS Control Tower allows you to customize new and existing AWS accounts when you provision their resources from the AWS Control Tower console. After you set up account factory customization, AWS Control Tower automates this process for future provisioning, so you don't have to maintain any pipelines. Customized accounts are available for use immediately after the resources are provisioned.

Your customized accounts are provisioned in account factory, through AWS CloudFormation templates, or with Terraform. You'll define a template that serves as customized account blueprint. Your blueprint describes the specific resources and configurations you require when an account is provisioned. Pre-defined blueprints, built and managed by AWS partners, also are available. For more information about partner-managed blueprints, see the Service Catalog Getting Started Library.

**Note**
AWS Control Tower contains proactive controls, which monitor AWS CloudFormation resources in AWS Control Tower. Optionally, you can activate these controls in your landing zone. When you apply proactive controls, they check to make sure that the resources you're about to deploy to your accounts are compliant with your organization's policies and procedures. For more information about proactive controls, see Proactive controls (p. 230).

Your account blueprints are stored in an AWS account, which for our purposes is referred to as a hub account. Blueprints are stored in the form of an Service Catalog product. We call this product a blueprint, to distinguish it from any other Service Catalog products. To learn more about how to create Service Catalog products, see Creating products in the Service Catalog Administrator Guide.

**Apply blueprints to existing accounts**

You can apply customized blueprints to existing accounts, also, by following the Update account steps in the AWS Control Tower console. For details, see Update the account in the console (p. 123).

**Before you begin**

Before you begin to create customized accounts with AWS Control Tower Account Factory, you must have an AWS Control Tower landing zone environment deployed, and you must have an organizational unit (OU) registered with AWS Control Tower, where your newly created accounts will be placed.

For more information about working with AFC, see Automate account customization using Account Factory Customization in AWS Control Tower.
Preparation for customization

- You may create a new account to serve as the hub account, or you may use an existing AWS account. We strongly recommend that you do not use the AWS Control Tower management account as your blueprint hub account.
- If you plan to enroll AWS accounts into AWS Control Tower and customize them, you must first add the AWSControlTowerExecution role to those accounts, as you would for any other account you are enrolling into AWS Control Tower.
- If you plan to use partner blueprints that have marketplace subscription requirements, you must configure these from your AWS Control Tower management account before you deploy the partner blueprints as account factory customization blueprints.

Topics

- Set up for customization (p. 130)
- Create a customized account from a blueprint (p. 134)
- Enroll and customize accounts (p. 135)
- Add a blueprint to an AWS Control Tower account (p. 135)
- Update a blueprint (p. 135)
- Remove a blueprint from an account (p. 136)
- Partner blueprints (p. 136)
- Considerations for Account Factory Customizations (AFC) (p. 136)
- In case of a blueprint error (p. 137)
- Customizing your policy document for AFC blueprints based on CloudFormation (p. 138)
- Additional permissions required for creating a Terraform-based Service Catalog product (p. 139)

Set up for customization

The next sections give steps to set up Account Factory for the customization process. We recommend that you set up delegated admin for the hub account, before you begin these steps.

Summary

- **Step 1. Create the required role.** Create an IAM role that grants permission for AWS Control Tower to have access to the (hub) account, where the Service Catalog products, also called blueprints, are stored.
- **Step 2. Create the Service Catalog product.** Create the Service Catalog product (also called a "blueprint product") that you’ll need for baselining the custom account.
- **Step 3. Review your custom blueprint.** Inspect the Service Catalog product (blueprint) that you created.
- **Step 4. Call your blueprint to create a customized account.** Enter the blueprint product information and the role information into the proper fields in Account Factory, in the AWS Control Tower console, while creating the account.

**Step 1. Create the required role**

Before you begin to customize accounts, you must set up a role that contains a trust relationship between AWS Control Tower and your hub account. When assumed, the role grants AWS Control Tower access to administer the hub account. The role must be named `AWSControlTowerBlueprintAccess`.
AWS Control Tower assumes this role to create a Portfolio resource on your behalf in Service Catalog, then to add your blueprint as a Service Catalog Product to this Portfolio, and then to share this Portfolio, and your blueprint, with your member account during account provisioning.

You'll create the AWSControlTowerBlueprintAccess role, as explained in the following sections.

Navigate to the IAM console to set up the required role.

To set up the role in an enrolled AWS Control Tower account

1. Federate or sign in as the principal in the AWS Control Tower management account.
2. From the federated principal in the management account, assume or switch roles to the AWSControlTowerExecution role in the enrolled AWS Control Tower account that you select to serve as the blueprint hub account.
3. From the AWSControlTowerExecution role in the enrolled AWS Control Tower account, create the AWSControlTowerBlueprintAccess role with proper permissions and trust relationships.

Note
To comply with AWS best practices guidance, it's important that you sign out of the AWSControlTowerExecution role immediately after you create the AWSControlTowerBlueprintAccess role.
To prevent unintended changes to resources, the AWSControlTowerExecution role is intended for use by AWS Control Tower only.

If your blueprint hub account isn't enrolled in AWS Control Tower, the AWSControlTowerExecution role won't exist in the account, and there's no need to assume it before you continue with setting up the AWSControlTowerBlueprintAccess role.

To set up the role in an unenrolled member account

1. Federate or sign in as a principal in the account that you wish to designate as the hub account, by means of your preferred method.
2. When signed in as the principal in the account, create the AWSControlTowerBlueprintAccess role with proper permissions and trust relationships.

The AWSControlTowerBlueprintAccess role must be set up to grant trust to two principals:

- The principal (user) that runs AWS Control Tower in the AWS Control Tower management account.
- The role named AWSControlTowerAdmin in the AWS Control Tower management account.

Here's an example trust policy, similar to one you will need to include for your role. This policy demonstrates the best practice of granting least-privilege access. When you make your own policy, replace the term YourManagementAccountId with the actual account ID of your AWS Control Tower management account, and replace the term YourControlTowerUserRole with the identifier of the IAM role for your management account.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": [
            "arn:aws:iam::YourManagementAccountId:role/service-role/AWSControlTowerAdmin",
```
"arn:aws:iam::YourManagementAccountId:role/YourControlTowerUserRole"
},
"Action": "sts:AssumeRole"
}]

**Required permissions policy**

AWS Control Tower requires that the managed policy named AWSServiceCatalogAdminFullAccess must be attached to the AWSControlTowerBlueprintAccess role. This policy provides permissions that Service Catalog looks for when it allows AWS Control Tower to administer your portfolio and Service Catalog Product resources. You can attach this policy when you're creating the role in the IAM console.

**Additional permissions may be required**

- If you store your blueprints in Amazon S3, AWS Control Tower also requires the AmazonS3ReadOnlyAccess permission policy for the AWSControlTowerBlueprintAccess role.
- The AWS Service Catalog Terraform type of product requires you to add some additional permissions to the AFC custom IAM policy, if you don’t utilize the default Admin policy. It requires these in addition to the permissions required to create the resources that you define in your terraform template.

**Step 2. Create the Service Catalog product**

To create an Service Catalog product, follow the steps at Creating products in the Service Catalog Administrator Guide. You'll add your account blueprint as a template when you create the Service Catalog product.

**Summary of steps to create a blueprint**

- Create or download an AWS CloudFormation template that will become your account blueprint. Some template examples are given later in this section.
- Sign in to the AWS account where you store your Account Factory blueprints (sometimes called the hub account).
- Navigate to the Service Catalog console. Choose Product list, and then choose Upload new product.
- In the Product details pane, enter details for your blueprint product, such as a name and description.
- Select Use a template file and then select Choose file. Select or paste the AWS CloudFormation template you've developed or downloaded for use as your blueprint.
- Choose Create product at the bottom of the console page.

You can download an AWS CloudFormation template from the Service Catalog reference architecture repository. One example from that repository helps set up a backup plan for your resources.

Here's an example template, for a fictitious company called Best Pets. It helps set up a connection to their pet database.

```
Resources:
  ConnectionStringGeneratorLambdaRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
```
Step 3. Review your custom blueprint

You can view your blueprint in the Service Catalog console. For more information, see Managing products in the Service Catalog Administrator Guide.

Step 4. Call your blueprint to create a customized account

When you follow the Create account workflow in the AWS Control Tower console, you'll see an optional section where you can enter information about the blueprint you'd like to use for customizing accounts.

Note

You must set up your customization hub account and add at least one blueprint (Service Catalog product) before you can enter that information into the AWS Control Tower console and begin to provision customized accounts.
Create or update a customized account in the AWS Control Tower console.

1. Enter the account ID for the account that contains your blueprints.
2. From that account, select an existing Service Catalog product; that is, select an existing blueprint.
3. Select the proper version of the blueprint (Service Catalog product), if you have more than one version.
4. (Optional) You can add or change a blueprint provisioning policy at this point in the process. The blueprint provisioning policy is written in JSON and attached to an IAM role, so it can provision the resources that are specified in the blueprint template. AWS Control Tower creates this role in the member account so that Service Catalog can deploy resources using AWS CloudFormation stack sets. The role is named AWSControlTower-BlueprintExecution-bp-xxxx. The AdministratorAccess policy is applied here by default.
5. Choose the AWS Region or Regions in which you wish to deploy accounts based on this blueprint.
6. If your blueprint contains parameters, you can enter the values for the parameters into additional fields in the AWS Control Tower workflow. The additional values may include: a GitHub repository name, a GitHub branch, an Amazon ECS cluster name, and a GitHub identity for the repository owner.
7. You can customize accounts at a later time by following the Account update process, if your hub account or blueprints are not yet ready.

For more details, see Create a customized account from a blueprint (p. 134).

Create a customized account from a blueprint

After you have created custom blueprints, you can start creating custom accounts in AWS Control Tower account factory.

Follow these steps to deploy a custom blueprint when you're creating a new AWS account:

1. Go to AWS Control Tower in the AWS Management Console.
2. Select Account factory and Create account.
3. Enter account details such as account name and email address.
4. Configure IAM Identity Center details with email address and user name.
5. Select a registered OU where your account will be added.
6. Expand the Account factory customization section.
7. Enter the account ID of the blueprint hub account that contains your Service Catalog products and choose Validate. For more information about a blueprint hub account, see Customize accounts with Account Factory Customization (AFC) (p. 129).
8. Select the dropdown menu that contains all blueprints from your Service Catalog Product List (all custom and partner blueprints). Choose a blueprint and corresponding version to deploy.
9. If your blueprint contains parameters, these fields are displayed for you to populate. Default values are pre-populated.
10. Finally, select where you'll deploy your blueprint, either Home Region or All governed Regions. Global resources such as Route 53 or IAM, may need to be deployed to a single Region only. Regional resources, such as Amazon EC2 instances or Amazon S3 buckets, could be deployed to all governed Regions.
11. After all fields are completed, select Create account.

Note
Blueprints created with Terraform can deploy to one Region only, not multiple Regions.
You can view the progress of your account provisioning on the **Organization** page. When your account provisioning is complete, the resources specified by your blueprint are already deployed within it. To view the details of the account and blueprint, go to the **Account details** page.

### Enroll and customize accounts

To enroll and customize accounts in the AWS Control Tower console.

1. Navigate to the AWS Control Tower console and select **Organization** from the left navigation.
2. You will see a list of your available accounts. Identify the account you would like to enroll with a custom blueprint. The **State** column for that account should reflect the account in a **Not enrolled** status.
3. Select the radio button to the left of the account and choose the **Actions** dropdown menu, in the top right of the screen. Here you will select the **Enroll** option.
4. Complete the **Access configuration** section with the account's IAM Identity Center information.
5. Select the registered OU where your account will become a member.
6. Complete the **Account factory customization** section using the same steps as 7-12 of the **Create account** procedure. For more information, see [Provision Account Factory accounts with AWS Service Catalog](#).

You can view the status of your account progress on the **Organization** page. When your account enrollment is complete, the resources specified by the blueprint are already deployed within it.

### Add a blueprint to an AWS Control Tower account

To add a blueprint to an existing AWS Control Tower member account, follow the **Update account** workflow in the AWS Control Tower console, and choose a new blueprint to add to the account. For more information, see [Update and move Account Factory accounts with AWS Control Tower or with AWS Service Catalog](#).

**Note**
If you add a new blueprint to an account, the existing blueprint is overwritten.

**Note**
One blueprint may be deployed per AWS Control Tower account.

### Update a blueprint

The following procedures describe how to update custom blueprints and how to deploy them.

#### To update your custom blueprints

1. Update your AWS CloudFormation template or your Terraform-based template (that is, your blueprint) with your new configurations.
2. Save the updated template (blueprint) as a new version in Service Catalog.

#### To deploy your updated blueprint

1. Navigate to the **Organization** page in the AWS Control Tower console.
2. Filter the **Organization** page by blueprint name and version.
3. Follow the **Update account** process, and deploy the latest blueprint version in your account.

**If a blueprint update is unsuccessful**
AWS Control Tower allows blueprint updates when the provisioned product is in the AVAILABLE state. If your provisioned product is in a TAINTED state, the update will fail. We recommend the following workaround:

1. In the AWS Service Catalog console, manually update the TAINTED provisioned product to change the state to AVAILABLE. For more information, see Updating provisioned products.
2. Then, follow the update account process from AWS Control Tower to fix the blueprint deployment error.

*We recommend this manual step because:* When you remove a blueprint, it can cause resources in the member account to be removed. Removing resources may affect your existing workloads. For this reason, we recommend this method rather than the alternative way of updating a blueprint—which is by removing and replacing the original blueprint—especially if you are running production workloads.

**Remove a blueprint from an account**

To remove a blueprint from an account, follow the Update account workflow to remove the blueprint and return the account to the AWS Control Tower default configurations.

As you enter the Update account workflow in the console, you will see that all of the account details are populated, and the customization details are not populated. If you leave these AFC details blank, AWS Control Tower removes the blueprint from the account. You will see a warning message before the action begins.

**Note**
AWS Control Tower adds a blueprint to an account only if you select a blueprint during the Create account or Update account process.

**Partner blueprints**

AWS Control Tower Account Factory Customization (AFC) provides access to pre-defined customization blueprints that are built and managed by AWS Partners. These partner blueprints help you customize your accounts for specific use cases. Each partner's blueprints help you build customized accounts, which are pre-configured to work with the product offerings from that particular partner.

To view a complete list of AWS Control Tower partner blueprints, navigate to the Service Catalog Getting Started Library in your console. Search for the source type AWS Control Tower Blueprints.

**Considerations for Account Factory Customizations (AFC)**

- AFC supports customization using a single Service Catalog blueprint product only.
- The Service Catalog blueprint products must be created in the hub account, and in the same Region as the AWS Control Tower landing zone home Region.
- The AWSControlTowerBlueprintAccess IAM role must be created with the proper name, permissions, and trust policy.
- AWS Control Tower supports two deployment options for blueprints: deploy to the home Region only, or deploy to all Regions governed by AWS Control Tower. Selection of Regions is not available.
- When you update a blueprint in a member account, the blueprint hub account ID and the Service Catalog blueprint product cannot be changed.
- AWS Control Tower doesn't support removing an existing blueprint and adding a new blueprint in a single blueprint update operation. You can remove a blueprint and then add a new blueprint in separate operations.
• AWS Control Tower changes behavior, based on whether you are creating or enrolling customized accounts, or non-customized accounts. If you are not creating or enrolling customized accounts with blueprints, AWS Control Tower creates an Account Factory provisioned product (through Service Catalog) in the AWS Control Tower management account. If you are specifying customization when creating or enrolling accounts with blueprints, AWS Control Tower does not create an Account Factory provisioned product in the AWS Control Tower management account.

In case of a blueprint error

Error while applying a blueprint

If an error occurs during the process of applying a blueprint to an account—either a new account or an existing account that you are enrolling into AWS Control Tower—the recovery procedure is the same. The account will exist, but it is not customized, and it is not enrolled into AWS Control Tower. To continue, follow the steps to enroll the account into AWS Control Tower, and add the blueprint at time of enrollment.

Error while creating the AWSControlTowerBlueprintAccess role, and workarounds

When you create the AWSControlTowerBlueprintAccess role from an AWS Control Tower account, you must be signed in as the principal using the AWSControlTowerExecution role. If you are signed in as any other, the CreateRole operation is prevented by an SCP, as shown in the artifact that follows:

```
{
  "Condition": {
    "ArnNotLike": {
      "aws:PrincipalArn": [
        "arn:aws:iam::*:role/AWSControlTowerExecution",
        "arn:aws:iam::*:role/stacksets-exec-*"
      ]
    },
    "Action": [
      "iam:AttachRolePolicy",
      "iam:CreateRole",
      "iam:DeleteRole",
      "iam:DeleteRolePermissionsBoundary",
      "iam:DeleteRolePolicy",
      "iam:DetachRolePolicy",
      "iam:PutRolePermissionsBoundary",
      "iam:PutRolePolicy",
      "iam:UpdateAssumeRolePolicy",
      "iam:UpdateRole",
      "iam:UpdateRoleDescription"
    ],
    "Resource": [
      "arn:aws:iam::*:role/aws-controltower-*",
      "arn:aws:iam::*:role/AWSControlTower*",
      "arn:aws:iam::*:role/stacksets-exec-*"
    ],
    "Effect": "Deny",
    "Sid": "GRIAMROLEPOLICY"
  }
}
```

The following workarounds are available:

• (Most recommended) Assume the AWSControlTowerExecution role and create the AWSControlTowerBlueprintAccess role. If you choose this workaround, be sure to sign out from the AWSControlTowerExecution role immediately afterward, to prevent unintended changes to resources.
• Sign into an account that is not enrolled in AWS Control Tower, and therefore not subject to this SCP.
• Temporarily edit this SCP to permit the operation.
• (Strongly not recommended) Use your AWS Control Tower management account as your hub account, so it is not subject to the SCP.

Customizing your policy document for AFC blueprints based on CloudFormation

When you enable a blueprint through account factory, AWS Control Tower directs AWS CloudFormation to create a StackSet on your behalf. AWS CloudFormation requires access to your managed account to create AWS CloudFormation stacks in the StackSet. Although AWS CloudFormation already has administrator privileges in the managed account through the AWSControlTowerExecution role, this role is not assumable by AWS CloudFormation.

As part of enabling a blueprint, AWS Control Tower creates a role in the member account, which AWS CloudFormation may assume to complete the StackSet management tasks. The simplest way to enable your customized blueprint through account factory is to use an allow-all policy, because those policies are compatible with any blueprint template.

However, best practices suggest that you must restrict the permissions for AWS CloudFormation in the target account. You can provide a customized policy, which AWS Control Tower applies to the role it creates for AWS CloudFormation to use. For example, if your blueprint creates an SSM Parameter called something-important, you could provide the following policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFormationActionsOnStacks",
      "Effect": "Allow",
      "Action": "cloudformation:*",
      "Resource": "arn:aws:cloudformation:*:*:stack/*"
    },
    {
      "Sid": "AllowSsmParameterActions",
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter",
        "ssm:DeleteParameter",
        "ssm:GetParameter",
        "ssm:GetParameters"
      ],
      "Resource": "arn:*:ssm:*:*:parameter/something-important"
    }
  ]
}
```

The AllowCloudFormationActionsOnStacks statement is required for all AFC custom policies; AWS CloudFormation uses this role to create stack instances, therefore it requires permission to perform AWS CloudFormation actions on stacks. The AllowSsmParameterActions section is specific to the template being enabled.

Resolve permission issues

When you enable a blueprint with a restricted policy, you may find that there are insufficient permissions to enable the blueprint. To resolve these issues, revise your policy document and update the member account’s blueprint preferences to use the corrected policy. To check that the policy is sufficient to enable
the blueprint, ensure that the AWS CloudFormation permissions are granted, and that you can create a stack directly using that role.

**Additional permissions required for creating a Terraform-based Service Catalog product**

When you're creating an Service Catalog Terraform product for AFC, Service Catalog requires certain permissions to be added to your AFC custom IAM policy, in addition to permissions required to create the resources defined in your template. If you choose the default full **Admin** policy, you do not need to add these extra permissions.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": ["resource-groups:CreateGroup",
                        "resource-groups:ListGroupResources",
                        "resource-groups:DeleteGroup",
                        "resource-groups:Tag"],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": ["tag:GetResources",
                        "tag:GetTagKeys",
                        "tag:GetTagValues",
                        "tag:TagResources",
                        "tag:UntagResources"],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": ["s3:GetObject"],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:ExistingObjectTag/servicecatalog:provisioning": "true"
                }
            }
        }
    ]
}
```

For more information about creating Terraform products in Service Catalog, see [Step 5: Create launch roles](#) in the Service Catalog Administrator Guide.

**Provision accounts with AWS Control Tower Account Factory for Terraform (AFT)**

AWS Control Tower Account Factory for Terraform (AFT) adopts a GitOps model that automates the process of account provisioning and updating in AWS Control Tower.
Prerequisites

Before getting started with AFT, you must create the following:

- A fully deployed AFT environment. For more information, see Overview of AWS Control Tower Account Factory for Terraform (AFT) and Deploy AWS Control Tower Account Factory for Terraform (AFT).
- One or more AFT git repositories in your fully deployed AFT environment. For more information, see Post-deployment steps for AFT.

Tip

Optionally, you can create an account template folder in the aft-account-customizations repository.

For information about AWS Regions where AFT has deployment limitations, see Control limitations (p. 28).

Provision a new account with AFT

To provision a new account with AFT, create an account request Terraform file. This file contains the input for parameters in the aft-account-request repository. After creating an account request Terraform file, begin processing your account request by running `git push`. This command invokes the ct-aft-account-request operation in the AWS CodePipeline, which is created in the AFT management account after account provisioning finishes. For more information, see AFT account provisioning pipeline.

Account request Terraform file parameters

You must include the following parameters in your account request Terraform file. You can view an example account request Terraform file on GitHub.

- The value of module name must be unique per the AWS account request.
- The value of module source is the path to the account request Terraform module that AFT provides.
- The value of control_tower_parameters captures the required input to create an AWS Control Tower account. The value includes the following input fields:
  - AccountEmail
  - AccountName
  - ManagedOrganizationalUnit
  - SSOUEmail
  - SSOUFirstName
  - SSOULastName

Note

The input that you provide for control_tower_parameters can't be changed during the account provisioning.

The supported formats for specifying ManagedOrganizationalUnit in the aft-account-request repository include OUName and OUName (OU-ID).
• account_tags captures user-defined keys and values, which can tag AWS accounts according to business criteria. For more information, see Tagging AWS Organizations resources in the AWS Organizations User Guide.

• The value of change_management_parameters captures additional information, such as why an account request was created and who initiated the account request. The value includes the following input fields:
  • change_reason
  • change_requested_by

• custom_fields captures additional metadata with keys and values that deploy as SSM parameters in the vended account under /aft/account-request/custom-fields/. You can reference this metadata during account customizations to deploy proper controls. For example, an account that's subject to regulatory compliance might deploy additional AWS Config Rules. The metadata that you collect with custom_fields can invoke additional processing during account provisioning and updating. If a custom field is removed from the account request, the custom field is removed from the SSM Parameter Store for the vended account.

• (Optional) account_customizations_name captures the account template folder in the aft-account-customizations repository. For more information, see Account customizations.

Submit multiple account requests

AFT processes account requests one at a time, but you can submit multiple account requests to the AFT pipeline. When you submit multiple account requests to the AFT pipeline, AFT queues and processes the account requests in a first-in, first-out order.

Note
You can create an account request Terraform file for each account that you want AFT to provision or cascade multiple account requests in a single account request Terraform file.

Update an existing account

You can update accounts that AFT provisions by editing previously submitted account requests and running git push. This command invokes the account provisioning workflow and can process account update requests. You can update the input for ManagedOrganizationalUnit, which is part of the required value for control_tower_parameters, and other parameters in the account request Terraform file. For more information, see Provision a new account with AFT.

Note
The input that you provide for control_tower_parameters can't be changed during account provisioning.

The supported formats for specifying ManagedOrganizationalUnit in the aft-account-request repository include OUName and OUName (OU-ID).

Update an account that AFT doesn't provision

You can update AWS Control Tower accounts created outside of AFT by specifying the account in the aft-account-request repository.

Note
Make sure that all account details are correct and consistent with the AWS Control Tower organization and respective AWS Service Catalog provisioned product.

Prerequisites for updating an existing AWS account with AFT

• The AWS account must be enrolled in AWS Control Tower.
• The AWS account must be part of the AWS Control Tower organization.
Deploy AWS Control Tower Account Factory for Terraform (AFT)

This section is for administrators of AWS Control Tower environments who wish to set up Account Factory for Terraform (AFT) in their existing environment. It describes how to set up an Account Factory for Terraform (AFT) environment with a new, dedicated AFT management account.

**Note**
A Terraform module deploys AFT. This module is available in the [AFT repository](https://github.com/) on GitHub, and the entire AFT repository is considered the module.
We recommend that you refer to the AFT modules on GitHub instead of cloning the AFT repository. This way you can control and consume updates to the modules as they are available.

For details about the latest releases of the AWS Control Tower Account Factory for Terraform (AFT) functionality, see the [Releases file](https://github.com/) for this GitHub repository.

**Deployment prerequisites**
Before you configure and launch your AFT environment, you must have the following:

- An AWS Control Tower landing zone. For more information, see [Plan your AWS Control Tower landing zone](https://aws.amazon.com/pt/aft/landingzones/).
- A home Region for your AWS Control Tower landing zone. For more information, see [How AWS Regions work with AWS Control Tower](https://aws.amazon.com/about-aws/whats-new/regions/).
- A Terraform version and distribution. For more information, see [Terraform and AFT versions](https://www.terraform.io/docs/).
- A VCS provider for tracking and managing changes to code and other files.

**Note**
If you'd like to choose a different VCS provider, see [Alternatives for version control of source code in AFT](https://aws.amazon.com/pt/aft/glossary/#versioncontrol).

- A runtime environment where you can run the Terraform module that installs AFT.
- AFT feature options. For more information, see [Enable feature options](https://aws.amazon.com/pt/aft/glossary/#features).

**Configure and launch your AWS Control Tower Account Factory for Terraform**

The following steps assume that you're familiar with the Terraform workflow. You can also learn more about deploying AFT by following the [Introduction to AFT](https://aws.amazon.com/pt/aft/lab/) lab on the AWS Workshop Studio website.

**Step 1: Launch your AWS Control Tower landing zone**

Complete the steps in [Getting started with AWS Control Tower](https://aws.amazon.com/pt/aft/guides/). This is where you create the AWS Control Tower management account and set up your AWS Control Tower landing zone.

**Note**
Make sure to create a role for the AWS Control Tower management account that has [AdministratorAccess](https://docs.aws.amazon.com/pt/AWSCloudFormation/latest/UserGuide/aws-properties-iam-group.html) credentials. For more information, see the following:

Step 2: Create a new organizational unit for AFT (recommended)

We recommend that you create a separate OU in your AWS organization. This is where you deploy the AFT management account. Create the new OU with your AWS Control Tower management account. For more information, see Create a new OU.

Step 3: Provision the AFT management account

AFT requires that you provision an AWS account dedicated to AFT management operations. The AWS Control Tower management account, which is associated to your AWS Control Tower landing zone, vends the AFT management account. For more information, see Provision accounts with AWS Service Catalog Account Factory.

**Note**
If you created a separate OU for AFT, make sure to select this OU when you create the AFT management account.

**Note**
You might wait up to 30 minutes before the AFT management account is fully provisioned.

Step 4: Verify the Terraform environment is available for deployment

This step assumes that you have experience with Terraform and have procedures in place for executing Terraform. For more information, see Command: init on the HashiCorp Developer website.

**Note**
AFT supports Terraform Version 0.15.x or later.

Step 5: Call the Account Factory for Terraform module to deploy AFT

Call the AFT module with the role that you created for the AWS Control Tower management account that has AdministratorAccess credentials. AWS Control Tower provisions a Terraform module through the AWS Control Tower management account, which establishes all of the infrastructure required to orchestrate AWS Control Tower Account Factory requests.

**Note**
You can view the AFT module in the AFT repository on GitHub. The entire GitHub repository is considered the AFT module. Refer to the README file for information about the input that's required to run the AFT module and deploy AFT. Alternatively, you can view the AFT module in the Terraform Registry.

If you have pipelines in your environment that are established for managing Terraform, you can integrate the AFT module into your existing workflow. Otherwise, run the AFT module from any environment that's authenticated with the required credentials.

A Terraform state file is generated when you deploy AFT. This artifact describes the state of the resources that Terraform created. If you plan to update the AFT version, make sure to preserve the Terraform state file, or set up a Terraform backend using Amazon S3 and DynamoDB. The AFT module doesn't manage a backend Terraform state.

**Note**
You're responsible for protecting the Terraform state file. Some input variables might contain sensitive values, such as a private ssh key or Terraform token. Depending on your deployment method, these values can be viewable as plain text in the Terraform state file. For more information, see Sensitive data in State on the HashiCorp website.

Timeout causes deployment to fail. We recommend using AWS Security Token Service (STS) credentials to ensure you have a timeout that's sufficient for a full deployment. The minimum timeout for AWS STS
credentials is 60 minutes. For more information, see Temporary security credentials in IAM in the AWS Identity and Access Management User Guide.

Note
You might wait up to 30 minutes for AFT to finish deploying through the Terraform module.

Post-deployment steps

After the AFT infrastructure deployment is complete, follow these additional steps to complete the setup process and get ready to provision accounts.

Step 1: (Optional) Complete AWS CodeStar Connections with your desired VCS provider

If you choose a third-party VCS provider, AFT establishes AWS CodeStar Connections, and you confirm them. Refer to Alternatives for version control of source code in AFT (p. 162) to learn how to set up AFT with your preferred VCS.

The initial step of establishing the AWS CodeStar connection is accomplished by AFT. You must confirm the connection.

Step 2: (Mandatory) Populate each repository

AFT requires that you manage four repositories:

1. Account requests – This repository handles placing or updating account requests. Examples available. For more information about AFT account requests, see Provision a new account with AFT (p. 140).
2. AFT account provisioning customizations – This repository manages customizations that are applied to all accounts created by and managed with AFT, before beginning the global customizations stage. Examples available. To create AFT account provisioning customizations, see Create your AFT account provisioning customizations state machine (p. 158).
3. Global customizations – This repository manages customizations that are applied to all accounts created by and managed with AFT. Examples available. To create AFT global customizations, see Apply global customizations (p. 159).
4. Account customizations – This repository manages customizations that are applied only to specific accounts created by and managed with AFT. Examples available. To create AFT account customizations, see Apply account customizations (p. 159).

AFT expects that each of these repositories follow a specific directory structure. The templates that are used to populate your repositories and instructions that describe how to populate the templates are available in the Account Factory for Terraform module in the AFT github repository.

Overview of AWS Control Tower Account Factory for Terraform (AFT)

Account Factory for Terraform (AFT) sets up a Terraform pipeline to help you provision and customize accounts in AWS Control Tower. AFT provides you with the advantage of Terraform-based account provisioning while allowing you to govern your accounts with AWS Control Tower.

With AFT you create an account request Terraform file to get the input that triggers the AFT workflow for account provisioning. After the account provisioning stage is complete, AFT automatically runs a series of steps before the account customizations stage begins. For more information, see AFT account provisioning pipeline.

AFT supports Terraform Cloud, Terraform Enterprise, and Terraform Open Source. With AFT you can initiate account creation using an input file and a simple git push command and customize new or
existing accounts. Account creation includes all of the AWS Control Tower governance benefits and account customizations that help you meet your organization's standard security procedures and compliance guidelines.

AFT supports account customization request tracing. Every time you submit an account customization request, AFT generates a unique tracing token that passes through an AFT customizations AWS Step Functions state machine, which logs the token as part of its execution. You can then use Amazon CloudWatch Logs insights queries to search timestamp ranges and retrieve the request token. As a result, you can see payloads that accompany the token, so you can trace your account customization request throughout the entire AFT workflow. For information about CloudWatch Logs and Step Functions, see the following:

- [What is Amazon CloudWatch Logs?](#) in the [Amazon CloudWatch Logs User Guide](#)
- [What is AWS Step Functions?](#) in the [AWS Step Functions Developer Guide](#)

AFT combines the capabilities of other AWS services as [Component services](#), to build a framework, with pipelines that deploy Terraform Infrastructure as Code (IaC). AFT enables you to:

- Submit account provisioning and update requests in a GitOps model
- Store account metadata and audit history
- Apply account-level tags
- Add customizations to all accounts, to a set of accounts, or to individual accounts
- Enable feature options

AFT creates a separate account, called the **AFT management account**, to deploy AFT capabilities. Before you can set up AFT, you must have an existing AWS Control Tower landing zone. The AFT management account is not the same as the AWS Control Tower management account.

**AFT offers flexibility**

- **Flexibility for your platform**: AFT supports any Terraform Distribution for initial deployment and ongoing operation: Open Source, Cloud, and Enterprise.
- **Flexibility for your version control system**: AFT natively relies on AWS CodeCommit, but it supports alternative sources for AWS CodeStar Connections.

**AFT offers feature options**

You can enable several feature options, based on best practices:

- Creating an organization-level CloudTrail for logging data events
- Deleting the AWS default VPC for accounts
- Enrolling provisioned accounts into the AWS Enterprise Support plan

**Note**

The AFT pipeline is not intended for use in deploying resources, such as Amazon EC2 instances, that your accounts require to run your applications. It is intended solely for automated provisioning and customizing of AWS Control Tower accounts.

**Video Walkthrough**

This video (7:33) describes how to deploy accounts with AWS Control Tower Account Factory for Terraform. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.
AFT Architecture

Order of operations

You run AFT operations in the AFT management account. For a full account provisioning workflow, the order of stages from left to right in the diagram are as follows:

1. Account requests are created and submitted to the pipeline. You can create and submit more than one account request at a time. Account Factory processes requests in a first-in-first-out order. For more information, see Submit multiple account requests.
2. Each account is provisioned. This stage runs in the AWS Control Tower management account.
3. Global customizations run in the pipelines that are created for each vended account.
4. If customizations are specified in the initial account provisioning requests, the customizations run only on targeted accounts. If you have an account that's already provisioned, you must initiate further customizations manually in the account's pipeline.

Cost

No additional charge exists for AFT. You pay only for the resources deployed by AFT, the AWS services enabled by AFT, and the resources you deploy in your AFT environment.

The default AFT configuration includes the allocation of AWS PrivateLink endpoints, for enhanced data protection and security, and a NAT gateway that is required to support AWS CodeBuild. For details on the pricing of this infrastructure, see the AWS PrivateLink pricing and the Amazon VPC pricing for the NAT Gateway. Contact your AWS account representative for more specific information about managing these costs. You can change these default settings for AFT.

Terraform and AFT versions

Account Factory for Terraform (AFT) supports Terraform version 0.15.x or later. You must provide a Terraform version as an input parameter for the AFT deployment process, as shown in the example that follows.

```
terraform_version = "0.15.1"
```
Terraform distributions

AFT supports three Terraform distributions:

• Terraform OSS
• Terraform Cloud
• Terraform Enterprise

These distributions are explained in the sections that follow. Provide the Terraform distribution of your choice as an input parameter during the AFT bootstrap process. For more information on AFT deployment and input parameters, see Deploy AWS Control Tower Account Factory for Terraform (AFT) (p. 142).

If you choose the Terraform Cloud or Terraform Enterprise distributions, the API token you specify for `terraform_token` must be a User or Team API token. An Organization token is not supported for all required APIs. For security reasons, you must avoid checking in this token's value to your version control system (VCS) by assigning a terraform variable, as shown in the example that follows.

```bash
# Sensitive variable managed in Terraform Cloud:
terraform_token = var.terraform_cloud_token
```

Terraform Open Source Software (Terraform OSS)

When you select Terraform OSS as your distribution, AFT manages the Terraform backend for you in the AFT management account. AFT downloads the `terraform-cli` of your specified Terraform version to run during the AFT deployment and the AFT pipeline phases. The resulting Terraform state configuration is stored in an Amazon S3 bucket, named with the following form:

```
aft-backend-[account_id]-primary-region
```

AFT also creates an Amazon S3 bucket that replicates your Terraform state configuration in another AWS Region, for disaster recovery purposes, named with the following form:

```
aft-backend-[account_id]-secondary-region
```

We recommend that you enable multi-factor authentication (MFA) for delete functions on these Terraform state Amazon S3 buckets. To learn more about Terraform OSS, see the Terraform documentation.

To select Terraform OSS as your distribution, provide the following input parameter:

```
terraform_distribution = "oss"
```

Terraform Cloud

When you select Terraform Cloud as your distribution, AFT creates workspaces for the following components in your Terraform Cloud organization, which initiates an API-driven workflow.

• Account request
• AFT customizations for accounts that AFT provisions
• Account customizations for accounts that AFT provisions
• Global customizations for accounts that AFT provisions
Terraform Cloud manages the resulting Terraform state configuration.

When you select Terraform Cloud as your distribution, provide the following input parameters:

- **terraform_distribution = "tfc"**
- **terraform_token** – This parameter contains the value of the Terraform Cloud token. AFT marks the value as sensitive and stores the value as a secure string in the SSM parameter store in the AFT management account. We recommend that you periodically rotate the value of the Terraform token according to your company's security policies and compliance guidelines. The Terraform token should be a User or Team level API token. Organization tokens are not supported.
- **terraform_org_name** – This parameter contains the name of your Terraform Cloud organization.

**Note**
Multiple AFT deployments in a single Terraform Cloud organization is not supported.

For information about how to set up Terraform Cloud, see the Terraform documentation.

**Terraform Enterprise**

When you select Terraform Enterprise as your distribution, AFT creates workspaces for the following components in your Terraform Enterprise organization, and it triggers API-driven workflow for the resulting Terraform runs.

- Account request
- AFT account provisioning customizations for accounts provisioned by AFT
- Account customizations for accounts provisioned by AFT
- Global customizations for accounts provisioned by AFT

The resulting Terraform state configuration is managed by your Terraform Enterprise setup.

To select Terraform Enterprise as your distribution, provide the following input parameters:

- **terraform_distribution = "tfe"**
- **terraform_token** – This parameter contains the value of your Terraform Enterprise token. AFT marks its value as sensitive and stores it as a secure string in the SSM parameter store, in the AFT management account. We recommend that you periodically rotate the value of the Terraform token, according to your company's security policies and compliance guidelines.
- **terraform_org_name** – This parameter contains the name of your Terraform Enterprise organization.
- **terraform_api_endpoint** – This parameter contains the URL of your Terraform Enterprise environment. The value of this parameter must be in the format:

  https://{fqdn}/api/v2/

See the Terraform documentation to learn more about how to set up Terraform Enterprise.

**Check the AFT version**

You can check your deployed AFT version by querying the AWS SSM Parameter Store key:

/af/t/config/aft/version

If you use the registry method, you can pin the version.
Enable feature options

AFT offers feature options based on best practices. You can opt-in to these features, by means of feature flags, during AFT deployment. Refer to Provision a new account with AFT (p. 140) for more information about AFT input configuration parameters.

These features are not enabled by default. You must explicitly enable each one in your environment.

Topics
- AWS CloudTrail data events (p. 149)
- AWS Enterprise Support plan (p. 150)
- Delete the AWS default VPC (p. 150)

AWS CloudTrail data events

When enabled, the AWS CloudTrail data events option configures these capabilities.

- Creates an Organization Trail in the AWS Control Tower management account, for CloudTrail
- Turns on logging for Amazon S3 and Lambda data events
- Encrypts and exports all the CloudTrail data events to an `aws-aft-logs-*` S3 bucket in the AWS Control Tower Log Archive account, with AWS KMS encryption
- Turns on the Log file validation setting

To enable this option, set the following feature flag to True in your AFT deployment input configuration.

```
aft_feature_cloudtrail_data_events
```

Prerequisite

Before you enable this feature option, be sure that trusted access for AWS CloudTrail is enabled in your organization.

To check the status of trusted access for CloudTrail:

1. Navigate to the AWS Organizations console.
2. Choose Services > CloudTrail.
3. Then select **Enable trusted access** in the upper right, if needed.

You may receive a warning message that advises you to use the AWS CloudTrail console, but in this case, disregard the warning. AFT creates the trail as part of enabling this feature option, after you allow trusted access. If trusted access is not enabled, you will receive an error message when AFT attempts to create your trail for data events.

**Note**

This setting works at the organization level. Enabling this setting affects all accounts in AWS Organizations, whether they are managed by AFT or not. All buckets in the AWS Control Tower Log Archive account at the time of enabling are excluded from Amazon S3 data events. Refer to the AWS CloudTrail User Guide to learn more about CloudTrail.

### AWS Enterprise Support plan

When this option is enabled, the AFT pipeline turns on the AWS Enterprise Support plan for accounts provisioned by AFT.

AWS accounts by default come with the AWS Basic Support plan enabled. AFT provides automated enrollment into the enterprise support level, for accounts that AFT provisions. The provisioning process opens a support ticket for the account, requesting it to be added to the AWS Enterprise Support plan.

To enable the Enterprise Support option, set the following feature flag to **True** in your AFT deployment input configuration.

```
aft_feature_enterprise_support=false
```

Refer to Compare AWS Support Plans to learn more about AWS Support Plans.

**Note**

To allow this feature to operate, you must enroll the payer account into the Enterprise Support plan.

### Delete the AWS default VPC

When you enable this option, AFT deletes all AWS default VPCs in the management account and in all AWS Regions, even if haven't deployed AWS Control Tower resources in those AWS Regions.

AFT doesn't delete AWS default VPCs automatically for any AWS Control Tower accounts that AFT provisions or for existing AWS accounts that you enroll in AWS Control Tower through AFT.

New AWS accounts are created with a VPC set up in each AWS Region, by default. Your enterprise may have standard practices for creating VPCs, which require you to delete the AWS default VPC and avoid enabling it, especially for the AFT management account.

To enable this option, set the following feature flag to **True** in your AFT deployment input configuration.

```
aft_feature_delete_default_vpcs_enabled
```

Refer to Default VPC and default subnets to learn more about default VPCs.

### Resource considerations for AWS Control Tower Account Factory for Terraform

When you set up your landing zone using AWS Control Tower Account Factory for Terraform, several types of AWS resources are created within your AWS accounts.
Search for resources

- You can use tags to search for the most updated list of AFT resources. The key-value pair for your search is:

  Key: managed_by | Value: AFT

- For component services that do not support tags, you can locate resources with a search for aft in the resource names.

Tables of resources initially created, by account

**AWS Control Tower Account Factory for Terraform management account**

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Identity and Access Management</td>
<td>Roles</td>
<td>AWSAFTAdministrator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSAFTExecution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSAFTService</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-ct-aft-*</td>
</tr>
<tr>
<td>AWS Identity and Access Management</td>
<td>Policies</td>
<td>aws-ct-aft-*</td>
</tr>
<tr>
<td>CodeCommit</td>
<td>Repositories</td>
<td>aws-ct-aft-*</td>
</tr>
<tr>
<td>CodeBuild</td>
<td>Build Projects</td>
<td>aws-ct-aft-*</td>
</tr>
<tr>
<td>Code Pipeline</td>
<td>Pipelines</td>
<td><em>-baseline-</em></td>
</tr>
<tr>
<td>Amazon S3</td>
<td>Buckets</td>
<td><em>-aws-ct-aft-</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-ct-aft-*</td>
</tr>
<tr>
<td>Lambda</td>
<td>Functions</td>
<td>aws-ct-aft-*</td>
</tr>
<tr>
<td>Lambda</td>
<td>Layers</td>
<td>aws-ct-aft-common-layer</td>
</tr>
<tr>
<td>DynamoDB</td>
<td>Tables</td>
<td>aws-ct-aft-request</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-ct-aft-request-audit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-ct-aft-request-metadata</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-ct-aft-controltower-events</td>
</tr>
<tr>
<td>Step Functions</td>
<td>State Machines</td>
<td>aws-ct-aft-prebaseline</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-ct-aft-prebaseline-customizations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-ct-aft-trigger-baseline</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-ct-aft-features</td>
</tr>
<tr>
<td>VPC</td>
<td>VPC</td>
<td>aws-ct-aft-vpc</td>
</tr>
<tr>
<td>Amazon SNS</td>
<td>Topics</td>
<td>aws-ct-aft-notifications</td>
</tr>
</tbody>
</table>
### AWS Control Tower management account

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Identity and Access Management</td>
<td>Roles</td>
<td>AWSAFTExecutionRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSAFTExecution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-ct-aft-controltower-events-rule</td>
</tr>
<tr>
<td>AWS Systems Manager</td>
<td>Parameter store</td>
<td>/aws-ct-aft/account/aws-ct-aft-management/account-id</td>
</tr>
<tr>
<td>AWS Organizations (Optional)</td>
<td>Service Control Policies</td>
<td>aws-ct-aft-protect-resources</td>
</tr>
<tr>
<td>CloudTrail (Optional)</td>
<td>Trails</td>
<td>aws-ct-aft-BaselineCloudTrail</td>
</tr>
<tr>
<td>AWS Support Center (Optional)</td>
<td>Support plans</td>
<td>Enterprise</td>
</tr>
</tbody>
</table>
AWS Control Tower log archive account

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Identity and Access Management</td>
<td>Roles</td>
<td>AWSAFTExecutionRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSAFTExecution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-ct-aft-cloudtrail-data-events-role</td>
</tr>
<tr>
<td>Key Management Service (KMS)</td>
<td>Customer Managed Keys</td>
<td>*-aws-ct-aft-kms-gd-findings</td>
</tr>
<tr>
<td>Amazon S3</td>
<td>Buckets</td>
<td><em>-aws-ct-aft-logs</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-ct-aft-s3-access-logs*</td>
</tr>
<tr>
<td>AWS Support Center (Optional)</td>
<td>Support plans</td>
<td>Enterprise</td>
</tr>
</tbody>
</table>

AWS Control Tower audit account

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Identity and Access Management</td>
<td>Roles</td>
<td>AWSAFTExecutionRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSAFTExecution</td>
</tr>
<tr>
<td>Amazon S3</td>
<td>Buckets</td>
<td><em>-aws-ct-aft-logs-</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>aws-ct-aft-s3-access-logs*</td>
</tr>
<tr>
<td>AWS Support Center (Optional)</td>
<td>Support plans</td>
<td>Enterprise</td>
</tr>
</tbody>
</table>

Required roles

In general, roles and policies are part of identity and access management (IAM) in AWS. Refer to the [AWS IAM User Guide](https://docs.aws.amazon.com/iam/latest/UserGuide/) for more information.

AFT creates multiple IAM roles and policies in the AFT management and AWS Control Tower management accounts to support the operations of the AFT pipeline. These roles are created based on the least privilege access model, which restricts permission to the minimally required sets of actions and resources for each role and policy. These roles and policies are assigned an AWS tag key:value pair, as managed_by:AFT for identification.

Besides these IAM roles, AFT creates three essential roles:

- the AWSAFTAdmin role
- the AWSAFTExecution role
- the AWSAFTService role

These roles are explained in the following sections.

**The AWSAFTAdmin role, explained**

When you deploy AFT, the AWSAFTAdmin role is created in the AFT management account. This role allows the AFT pipeline to assume the AWSAFTExecution role in AWS Control Tower and AFT provisioned accounts, thereby to perform actions related to account provisioning and customizations.
Here is the inline policy (JSON artifact) attached to the AWSAFTAdmin role:

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": [
                "arn:aws:iam::*:role/AWSAFTExecution",
                "arn:aws:iam::*:role/AWSAFTService"
            ]
        }
    ]
}
```

The following JSON artifact shows the trust relationship for the AWSAFTAdmin role. The placeholder number 012345678901 is replaced by the AFT management account ID number.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::012345678901:root"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

The AWSAFTExecution role, explained

When you deploy AFT, the AWSAFTExecution role is created in the AFT management and AWS Control Tower management accounts. Later, the AFT pipeline creates the AWSAFTExecution role in each AFT provisioned account during the AFT account provisioning stage.

AFT utilizes the AWSControlTowerExecution role initially, to create the AWSAFTExecution role in specified accounts. The AWSAFTExecution role allows the AFT pipeline to run the steps that are performed during the AFT framework's provisioning and provisioning customizations stages, for AFT provisioned accounts and for shared accounts.

**Distinct roles help you limit scope**

As a best practice, keep the customization permissions separate from the permissions allowed during your initial deployment of resources. Remember that the AWSAFTService role is intended for account provisioning, and the AWSAFTExecution role is intended for account customization. This separation limits the scope of permissions that are allowed during each phase of the pipeline. This distinction is especially important if you are customizing the AWS Control Tower shared accounts, because the shared accounts may contain sensitive information, such as billing details or user information.

Permissions for AWSAFTExecution role: **AdministratorAccess** – an AWS managed policy

The following JSON artifact shows the IAM policy (trust relationship) attached to the AWSAFTExecution role. The placeholder number 012345678901 is replaced by the AFT management account ID number.

```
{
}
```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::012345678901:role/AWSAFTAdmin"
    },
    "Action": "sts:AssumeRole"
  }
]
}

The AWSAFTService role, explained

The AWSAFTService role deploys AFT resources in all enrolled and managed accounts, including the shared accounts and management account. Resources formerly were deployed by the AWSAFTExecution role only.

The AWSAFTService role is intended for use by the service infrastructure to deploy resources during the provisioning stage, and the AWSAFTExecution role is intended to be used only to deploy customizations. By assuming the roles in this way, you can maintain more granular access control during each stage.

Permissions for AWSAFTService role: AdministratorAccess – an AWS managed policy

The following JSON artifact shows the IAM policy (trust relationship) attached to the AWSAFTService role. The placeholder number 012345678901 is replaced by the AFT management account ID number.

Trust policy for AWSAFTService

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::012345678901:role/AWSAFTAdmin"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

Component services

When you deploy AFT, components are added to your AWS environment from each of these AWS services.

- **AWS Control Tower** – AFT uses AWS Control Tower Account Factory in the AWS Control Tower management account to provision accounts.

- **Amazon DynamoDB** – AFT creates Amazon DynamoDB tables in the AFT management account, which store account requests, audit history of account updates, account metadata, and AWS Control Tower lifecycle events. AFT also creates DynamoDB Lambda triggers to initiate downstream processes, such as starting the AFT account provisioning workflow.

- **Amazon Simple Storage Service** – AFT creates Amazon Simple Storage Service (S3) buckets in the AFT management account and the AWS Control Tower log archive account, which store logs generated by the AWS services that the AFT pipeline requires. AFT also creates a Terraform backend S3 bucket, in primary and secondary AWS Regions, to store Terraform states generated during AFT pipeline workflows.
AFT account provisioning pipeline

After the account provisioning stage of the pipeline is complete, the AFT framework continues. It automatically runs a series of steps to ensure that the newly provisioned accounts have details in place, before the Account customizations (p. 158) stage begins.
Here are the next steps that the AFT pipeline runs.

1. Validates the account request input.
2. Retrieves information about the account provisioned, for example, the account ID.
3. Stores the account metadata in a DynamoDB table in the AFT management account.
4. Creates the AWSAFTExecution IAM role in the newly provisioned account. AFT assumes this role to perform the account customizations stage, because this role grants access to the account factory portfolio.
5. Applies the account tags that you provided as part of the account request input parameters.
6. Applies the AFT feature options you chose at the time of AFT deployment.
7. Applies the AFT account provisioning customizations you provided. The next section tells more about how to set up these customizations with an AWS Step Functions state machine, in a git repository. This stage is sometimes referred to as the account provisioning framework stage. It is part of the core provisioning process, but you've previously set up a framework that delivers customized integrations as part of your account provisioning workflow, before additional customizations are added to the accounts in the next stage.
8. For each account provisioned, it creates an AWS CodePipeline in the AFT management account, which will run to perform the (next, global) Account customizations (p. 158) stage.
9. Invokes the account customizations pipeline for each account provisioned (and targeted).
10. Sends a success or failure notification to the SNS topic, from which you can retrieve the messages.

Set up the account provisioning framework customizations with a state machine

If you set up custom, non-Terraform integrations before you provision your accounts, these customizations are included in your AFT account provisioning workflow. For example, you may require certain customizations to ensure that all accounts created by AFT are compliant with the standards and policies of your organization, such as security standards, and these standards may be added to accounts before additional customization. These account provisioning framework customizations are implemented on every provisioned account, before the global account customization stage begins next.

**Note**

The AFT feature described in this section is intended for advanced users who understand the functioning of AWS Step Functions. As an alternative, we recommend that you work with the global helpers in the account customizations stage.

The AFT account provisioning framework calls an AWS Step Functions state machine, which you define, to implement your customizations. Refer to the AWS Step Functions documentation to learn more about the possible state machine integrations.

Here are some common integrations.

- AWS Lambda functions in the language of your choice
- AWS ECS or AWS Fargate tasks, using Docker containers
- AWS Step Functions activities using custom workers, hosted either in AWS or on-premises
- Amazon SNS or SQS integrations

If no AWS Step Functions state machine is defined, the stage passes with a no-op. To create an AFT account provisioning customizations state machine, follow the instructions in Create your AFT account provisioning customizations state machine (p. 158). Before you add customizations, be sure you have the prerequisites in place.
These types of integrations are not part of AWS Control Tower, and they cannot be added during the global pre-API stage of AFT account customization. Instead, the AFT pipeline allows you to set up these customizations as part of the provisioning process, and they are run in the provisioning workflow. You must implement these customizations by creating your state machine ahead of time, before you kick off the AFT account provisioning stage, as described in the following sections.

Prerequisites for creating a state machine

- A fully deployed AFT. See Deploy AWS Control Tower Account Factory for Terraform (AFT) (p. 142) for more information about AFT deployment.
- Set up a git repository in your environment for AFT account provisioning customizations. See Post-deployment steps (p. 144) for more information.

Create your AFT account provisioning customizations state machine

Step 1: Modify the state machine definition

Modify the example customizations.asl.json state machine definition. The example is available in the git repository you set up for storing AFT account provisioning customizations, in your post-deployment steps. Refer to the AWS Step Functions Developer Guide to learn more about state machine definitions.

Step 2: Include the corresponding Terraform configuration

Include Terraform files with the .tf extension in the same git repository with the state machine definition for your custom integration. For example, if you choose to call a Lambda function in your state machine task definition, you'd include the lambda.tf file in the same directory. Make sure you include the required IAM roles and permissions for your custom configurations.

When you provide the appropriate input, the AFT pipeline automatically invokes your state machine and deploys your customizations as part of the AFT account provisioning framework stage.

To re-start the AFT account provisioning framework and customizations

AFT runs the account provisioning framework and customizations steps for every account vended through the AFT pipeline. To re-start account provisioning customizations, you can use one of these two methods:

1. Make any change to an existing account in the account request repo.
2. Provision a new account with AFT.

Account customizations

AFT can deploy standard or customized configurations in provisioned accounts. In the AFT management account, AFT provides one pipeline for each account. With this pipeline, you can implement your customizations in all accounts, in a set of accounts, or in individual accounts. You can run Python scripts, bash scripts, and Terraform configurations, or you can interact with the AWS CLI as part of your account customizations stage.

Overview

After your customizations are specified in your chosen git repositories, either the one where you store your global customizations or where you store your account customizations, the account customizations
stage is completed automatically by the AFT pipeline. To customize accounts retroactively, see Re-invoke customizations (p. 160).

Global customizations (optional)

You can choose to apply certain customizations to all accounts that are provisioned by AFT. For example, if you need to create a particular IAM role, or to deploy a custom control in every account, the global customizations stage in AFT pipeline allows you to do so, automatically.

Account customizations (optional)

To customize an individual account, or a set of accounts, differently than other AFT provisioned accounts, you can leverage the account customizations portion of the AFT pipeline to implement account-specific configurations. For example, only a certain account may require access to an internet gateway.

Customization prerequisites

Before you begin to customize accounts, be sure these prerequisites are in place.

- A fully deployed AFT. For information about how to deploy, see Configure and launch your AWS Control Tower Account Factory for Terraform (p. 142).
- Pre-populated git repositories for global customizations and account customizations in your environment. See Step 3: Populate each repository in Post-deployment steps (p. 144) for more information.

Apply global customizations

To apply global customizations, you must push a specific folder structure to your chosen repository.

- If your custom configurations are in the form of Python programs or scripts, place those under api_helpers/python folder in your repository.
- If your custom configurations are in the form of Bash scripts, place those under api_helpers folder in your repository.
- If your custom configurations are in the form of Terraform, place those under the terraform folder in your repository.
- Refer to the global customizations README file for more details on creating custom configurations.

Note

Global customizations are applied automatically, after the AFT account provisioning framework stage in the AFT pipeline.

Apply account customizations

You can apply account customizations by pushing a specific folder structure to your chosen repository. Account customizations are applied automatically in the AFT pipeline and after the global customizations stage. You can also create multiple folders that contain different account customizations in your account customizations repository. For each account customization that you require, use the following steps.

To apply account customizations

1. Step 1: Create a folder for an account customization
In your chosen repository, copy the ACCOUNT_TEMPLATE folder that AFT provides to a new folder. The name of your new folder should match the account_customizations_name that you provide in your account request.

2. **Add the configurations to your specific account customizations folder**

You can add configurations to your account customizations folder based on the format of your configurations.

- If your custom configurations are in the form of Python programs or scripts, place them under the `[account_customizations_name]/api_helpers/python` folder that's in your repository.
- If your custom configurations are in the form of Bash scripts, place them under the `[account_customizations_name]/api_helpers` folder that's in your repository.
- If your custom configurations are in the form of Terraform, place them under the `[account_customizations_name]/terraform` folder that's in your repository.

For more information about creating custom configurations, refer to the account customizations README file.

3. **Refer to the specific account_customizations_name parameter in the account request file**

The AFT account request file includes the input parameter `account_customizations_name`. Enter the name of your account customization as the value for this parameter.

**Note**

You can submit multiple account requests for accounts in your environment. When you want to apply different or similar account customizations, specify the account customizations using the `account_customizations_name` input parameter in your account requests. For more information, see [Submit multiple account requests](#).

**Re-invoke customizations**

AFT provides a way to re-invoke customizations in the AFT pipeline. This method is useful when you've added a new customization step, or when you are making changes to an existing customization. When you re-invoke, AFT initiates the customizations pipeline to make changes to the AFT provisioned account. An event-source-based re-invoke allows you to apply customizations to individual accounts, to all accounts, to accounts according to their OU, or to accounts selected according to tags.

Follow these three steps to re-invoke customizations for AFT-provisioned accounts.

**Step 1: Push changes to global or account customizations git repositories**

You can update your global and account customizations as needed and push changes back to your git repositories. At this point, nothing happens. The customizations pipeline must be invoked by an event source, as explained in the next two steps.

**Step 2: Start an AWS Step Function run for re-invoking customizations**

AFT provides an AWS Step Function called `aft-invoke-customizations` in the AFT management account. The purpose of that function is to re-invoke the customization pipeline for AFT-provisioned accounts.

Here is an example of an event schema (JSON format) you can create to pass input to the `aft-invoke-customizations` AWS Step Function.
The example event schema shows that you can choose accounts to include or exclude from the re-invoke process. You can filter by organizational unit (OU), account tags, and account ID. If you don't apply any filters and include the statement "type": "all", the customization for all AFT-provisioned accounts is re-invoked.

**Note**
If your version of AWS Control Tower is 1.6.5 or later, you can target nested OUs with the syntax OU Name (ou-id-1234). For more information, see the following topic on GitHub.

After you fill out the event parameters, Step Functions runs and invokes the corresponding customizations. AFT can invoke a maximum of 5 customizations at a time. Step Functions waits and loops until all accounts matching the event criteria are complete.

**Step 3: Monitor the AWS Step Function output and watch AWS CodePipeline running**

- The resulting Step Function output contains account IDs that match the Step Function input event source.
- Navigate to AWS CodePipeline under Developer Tools and view the corresponding customization pipelines for the account ID.

**Troubleshooting with AFT account customization request tracing**

Account customization workflows that are based on AWS Lambda emit logs containing target account and customization request IDs. AFT allows you to trace and troubleshoot customization requests with...
Amazon CloudWatch Logs by providing you with CloudWatch Logs Insights queries that you can use to filter CloudWatch Logs related to your customization request by your target account or customization request ID. For more information, see Analyzing log data with Amazon CloudWatch Logs in the Amazon CloudWatch Logs User Guide.

To use CloudWatch Logs Insights for AFT

2. From the navigation pane, choose Logs, and then choose Logs insights.
3. Choose Queries.
4. Under Sample queries, choose Account Factory for Terraform, and then select one of the following queries:
   - **Customization Logs by Account ID**
     
     Note
     
     Make sure to replace "YOUR-ACCOUNT-ID" with your target account ID.

     ```
     fields @timestamp, log_message.account_id as target_account_id, 
     log_message.customization_request_id as customization_request_id, log_message.detail as detail, logStream
     | sort @timestamp desc
     | filter log_message.account_id == "YOUR-ACCOUNT-ID" and @message like /
     customization_request_id/ 
     ```

   - **Customization Logs by Customization Request ID**
     
     Note
     
     Make sure to replace "YOUR-CUSTOMIZATION-REQUEST-ID" with your customization request ID. You can find your customization request ID in the output of the AFT account provisioning framework AWS Step Functions state machine. For more information about the AFT account provisioning framework, see AFT account provisioning pipeline

     ```
     fields @timestamp, log_message.account_id as target_account_id, 
     log_message.customization_request_id as customization_request_id, log_message.detail as detail, logStream
     | sort @timestamp desc 
     | filter log_message.customization_request_id == "YOUR-CUSTOMIZATION-REQUEST-ID" 
     ```

5. After you select a query, make sure to select a time interval, and then choose Run query.

Alternatives for version control of source code in AFT

AFT natively uses AWS CodeCommit for a source code version control system (VCS), but it allows other AWS CodeStar Connections that meet your business requirements or existing architecture. You can specify a third-party VCS as part of the AFT deployment prerequisites.

AFT supports the following source code control alternatives:

- GitHub
- GitHub Enterprise Server
- BitBucket

If you select AWS CodeCommit as your VCS, no additional steps are required. By default, AFT creates the necessary git repositories in your environment, with default names. However, you can override the default repository names for CodeCommit, as needed, to comply with your organizational standards.
Set up an alternative source code version control system (custom VCS) with AFT

To set up an alternative source code version control system for your AFT deployment, follow these steps.

**Step 1: Create git repositories in a supported third-party version control system (VCS).**

If you are not using AWS CodeCommit, you must create git repositories in your AFT-supported, third-party VCS provider environment for the following items.

- **AFT account requests.** [Sample code available](#). For more information about AFT account requests, see [Provision a new account with AFT](#).
- **AFT account provisioning customizations.** [Sample code available](#). For more information on AFT account provisioning customizations, see [Create your AFT account provisioning customizations state machine](#).
- **AFT global customizations.** [Sample code available](#). For more information on AFT global customizations, see [Account customizations](#).
- **AFT account customizations.** [Sample code available](#). For more information on AFT account customizations, see [Account customizations](#).

**Step 2: Specify the VCS configuration parameters required for AFT deployment**

The following input parameters are needed to configure your VCS provider as part of the AFT deployment.

- **vcs_provider:** If you are not using AWS CodeCommit, specify the VCS provider as "bitbucket", "github", or "githubenterprise", based on your use case.
- **github_enterprise_url:** For GitHub Enterprise customers only, specify the GitHub URL.
- **account_request_repo_name:** By default, this value is set to `aft-account-request` for AWS CodeCommit users. If you created your repository with a new name in CodeCommit or in an AFT-supported, third-party VCS provider environment, update this input value with your actual repository name. For BitBucket, Github, and GitHub Enterprise, the repository name must have the format `[Org]/[Repo]`.
- **account_customizations_repo_name:** By default, this value is set to `account-customizations` for AWS CodeCommit users. If you created your repository with a new name in CodeCommit or in an AFT-supported, third-party VCS provider environment, update this input value with your repository name. For BitBucket, Github, and GitHub Enterprise, the repository name must have the format `[Org]/[Repo]`.
- **account_provisioning_customizations_repo_name:** By default, this value is set to `account-provisioning-customizations` for AWS CodeCommit users. If you created your repository with a new name in CodeCommit or in an AFT-supported, third-party VCS provider environment, update this input value with your repository name. For BitBucket, Github, and GitHub Enterprise, the repository name must have the format `[Org]/[Repo]`.
- **global_customizations_repo_name:** By default, this value is set to `global-customizations` for AWS CodeCommit users. If you created your repository with a new name in CodeCommit or in an AFT-supported, third-party VCS provider environment, update this input value with your repository name. For BitBucket, Github, and GitHub Enterprise, the repository name must have the format `[Org]/[Repo]`.
- **account_request_repo_branch:** The branch is `main` by default, but the value can be overridden.

By default, AFT sources from the main branch of each git repository. You can override the branch name value with an additional input parameter. For more information about input parameters, refer to the README file in the [AFT Terraform module](#).
Step 3: Complete the AWS CodeStar connection for third-party VCS providers

When your deployment runs, AFT either creates the required AWS CodeCommit repositories, or it creates an AWS CodeStar connection for your chosen third-party VCS provider. In case of the latter, you must manually sign in to the AFT management account's console to complete the pending AWS CodeStar connection. See the AWS CodeStar documentation for further instructions on completing the AWS CodeStar connection.

Data protection

The AWS shared responsibility model applies to data protection in AFT. For data protection purposes, we recommend the following best practices for security.

- Follow the Data Protection guidelines provided by AWS Control Tower. For details, see Data Protection in AWS Control Tower (p. 1386).
- Preserve Terraform state configuration generated at the time of AFT deployment. For details, see Deploy AWS Control Tower Account Factory for Terraform (AFT) (p. 142).
- Rotate sensitive credentials periodically as directed by your organization's security policy. Examples of secrets are Terraform tokens, git tokens, and so forth.

Encryption at rest

AFT creates Amazon S3 buckets, Amazon SNS topics, Amazon SQS queues, and Amazon DynamoDB databases that are encrypted at rest with AWS Key Management Service keys. KMS keys created by AFT have yearly rotation enabled by default. If you choose the Terraform Cloud or Terraform Enterprise distributions of Terraform, AFT includes a AWS Systems Manager SecureString parameter to store Terraform token values that are sensitive.

AFT uses AWS services described in Component services (p. 155) that are, by default, encrypted at rest. For details, see the AWS documentation for each component AWS service of AFT, and learn about the data protection practices followed by each service.

Encryption in transit

AFT relies upon AWS services described in Component services (p. 155) that employ encryption in transit, by default. For details, see the AWS documentation for each component AWS service of AFT, and learn about the data protection practices followed by each service.

For Terraform Cloud or Terraform Enterprise distributions, AFT calls an HTTPS endpoint API for access to your Terraform organization. If you choose a third-party VCS provider supported by AWS CodeStar connections, AFT calls an HTTPS endpoint API for access to your VCS provider organization.

Remove an account from AFT

This topic describes how to remove an account from AFT, so the AFT pipeline stops deploying and updating the account.

Important
Removing an account from the AFT pipeline is irreversible and can result in a loss of state.

You might remove an account from AFT when you want to close an account for a retired application, isolate a compromised account, or move an account from one organization to another organization.

Note
Removing an account from AFT is different than deleting an AWS Control Tower account or AWS account. When you remove an account from AFT, AWS Control Tower still manages the account. To delete an AWS Control Tower account or AWS account, see the following:
To remove an account from the AFT pipelines

The following procedure describes how to remove an account from AFT.

1. **Remove account from git repository that stores account requests**

   In the git repository where you store account requests, delete the account request for the account you want to remove from AFT.

   When you remove an account request from the account request repository, AFT deletes the customization pipeline and account metadata. For more information, see the [1.8.0 release notes](https://github.com/aws/Account-Factory-for-Terraform/releases) for AFT on GitHub.

2. **Delete Terraform workspace (For Terraform Cloud and Terraform Enterprise customers only)**

   Delete the global customizations and account customizations workspaces for the account that you want to remove from AFT.

3. **Delete Terraform state from Amazon S3 backend**

   In the AFT management account, delete all relevant folders inside of the Amazon S3 buckets for the account that you want to remove from AFT.

   **Tip**

   In the following examples, replace `012345678901` with the AFT management account ID number.

   **Example: Terraform OSS**

   When you choose Terraform OSS, you find 3 folders for each account in the `aft-backend-012345678901-primary-region` and `aft-backend-012345678901-secondary-region` Amazon S3 buckets. These folders are related to the `account customizations state, customizations pipeline state, and global customizations state`.

   **Example: Terraform Cloud or Terraform Enterprise**

   When you choose Terraform Cloud or Terraform Enterprise, you find a folder for each account in the `aft-backend-012345678901-primary-region` and `aft-backend-012345678901-secondary-region` Amazon S3 buckets. These folders are related to the `customizations pipeline state`.

**Operational metrics**

By default, *Account Factory for Terraform (AFT)* sends anonymous operational metrics to AWS. We use this data to understand how customers are using AFT so we can improve the quality and features of the solution. You can opt out of data collection by changing a parameter during AFT deployment. When collection is enabled, the following data is sent to AWS:

- **Solution**: The AFT-specific identifier
- **Version**: The version of AFT
- **Universally Unique Identifier (UUID)**: Randomly generated, unique identifier for each AFT deployment
- **Timestamp**: Data-collection timestamp
- **Data**: AFT configuration and actions taken by the customer
AWS owns the data collected. Data collection is subject to the AWS Privacy Policy.

**Note**
Versions of AFT prior to 1.6.0 do not report usage metrics to AWS.

To opt out of reporting metrics:

- Set the input value of `aft_metrics_reporting` to `false` in your Terraform input configuration file, as shown in the example that follows, and redeploy AFT. This value is set to `true` by default, if you do not set it explicitly.

If you copy the example, remember to substitute your actual ID and Region values for the items given in strings with `x`.

```terraform
module "control_tower_account_factory" {
  source = "aws-ia/control_tower_account_factory/aws"

  # Required Vars
  ct_management_account_id    = "xxxxxxxxxxx"
  log_archive_account_id      = "xxxxxxxxxxx"
  audit_account_id            = "xxxxxxxxxxx"
  aft_management_account_id   = "xxxxxxxxxxx"
  ct_home_region              = "xx-xxxx-x"
  tf_backend_secondary_region = "xx-xxxx-x"

  # Optional Vars
  aft_metrics_reporting = false    # to opt out, set this value to false
}
```

**Account Factory for Terraform (AFT) troubleshooting guide**

This section can help you troubleshoot common issues that you might encounter when using Account Factory for Terraform (AFT).

**Topics**

- [General issues](#)
- [Issues related to account provisioning/registration](#)
- [Issues related to customizations invocation](#)
- [Issues related to the account customizations workflow](#)

**General issues**

- [Exceeded AWS resource quotas](#)

If your log groups indicate that you exceeded AWS resource quotas, contact AWS Support. Account Factory uses AWS services with resource quotas that include AWS CodeBuild, AWS Organizations, and AWS Systems Manager. For more information, see the following:

- [What is AWS CodeBuild?](#) in the CodeBuild User Guide.
- [What is AWS Organizations?](#) in the Organizations User Guide.
- [What is AWS Systems Manager?](#) in the Systems Manager User Guide.
- [Outdated version of Account Factory](#)
If you encounter an issue and believe the issue is a bug, make sure that you have the latest version of Account Factory. For more information, see [Updating the Account Factory version](#).

- **Local changes were made to the Account Factory source code**

  Account Factory is an open source project. AWS Control Tower supports the Account Factory core code. If you make a local change to the Account Factory core code, AWS Control Tower only supports your Account Factory deployment on a best-effort basis.

- **Insufficient Account Factory role permissions**

  Account Factory creates IAM roles and policies to manage vended account deployments and customizations. If you change these roles or policies, the Account Factory pipeline may be unable to perform certain actions. For more information, see [Required roles](#).

- **Account repositories not populated correctly**

  Make sure that you follow the [post-deployment steps](#) before provisioning accounts.

- **Not detecting drift after changing the OU manually**

  **Note**

  AWS Control Tower detects drift automatically. For information about resolving drift, see [Detect and resolve drift in AWS Control Tower](#).

  Drift isn’t detected when the organizational unit (OU) is changed manually. This is due to the event-driven nature of Account Factory. When an account request is submitted, the resource that Terraform manages is an Amazon DynamoDB item, not a direct account. After an item is changed, the request is put in a queue, where AWS Control Tower processes them through Service Catalog (the service that manages account details). If you change the OU manually, drift isn’t detected because the account request hasn’t changed.

### Issues related to account provisioning/registration

- **Account request (email address/name) already exists**

  The issue typically results in an Service Catalog product failure during provisioning or as `ConditionalCheckFailedException`.

  You can find more information about the issue by doing one of the following:

  - Review your Terraform or CloudWatch Logs log groups.
  - Review the failures that are emitted to the Amazon SNS topic `aft-failure-notifications`.

- **Malformed account request**

  Make sure that your account request follows the expected schema. For examples, see [terraform-aws-control_tower_account_factory](#) on GitHub.

- **Exceeded AWS Organizations resource quotas**

  Make sure that your account request doesn’t exceed AWS Organizations resource quotas. For more information, see [Quotas for AWS Organizations](#).

### Issues related to customizations invocation

- **Target account not onboarded to Account Factory**

  Make sure all accounts that are included in a customization request have been onboarded to Account Factory. For more information, see [Update an existing account](#).
• **Account that customization request targets exists in the DynamoDB table `aft-request-metadata`, but not in account request repository**

Format your customization invocation request to exclude the offending account by doing one of the following:

- In the DynamoDB table `aft-request-metadata`, delete the entry referencing the account that's no longer in your account request repository.
- Not using "all" as the target.
- Not targeting the OU that the account belongs to.
- Not targeting the account directly.

• **Used incorrect token for Terraform Cloud**

Make sure that you set up the correct token. Terraform Cloud only supports team-based tokens, not organization-based tokens.

• **Failed to create account before account customizations pipeline is created; can't customize account**

Make a change to the account specification in the account request repository. When you make a change, such as changing a tag value for an account, Account Factory follows the path that tries to create the pipeline, even if the pipeline doesn't exist.

### Issues related to the account customizations workflow

If you’re experiencing issues related to the account customizations workflow, make sure that your version of AFT is 1.8.0 or higher, and that you delete all instances of account-related metadata from your DynamoDB request table.

For information about AFT version 1.8.0, see [Release 1.8.0 on GitHub](https://github.com/aws/aws-account-factory/releases/tag/v1.8.0).

For information about how to check and update your version of AFT, see the following:

- [Check the AFT version](#)
- [Update the AFT version](#)

You can also trace and troubleshoot customization requests by using Amazon CloudWatch Logs Insights queries to filter logs containing your target account and customization request IDs. For more information, see [Troubleshooting with AFT account customization request tracing](#).
Detect and resolve drift in AWS Control Tower

Identifying and resolving drift is a regular operations task for AWS Control Tower management account administrators. Resolving drift helps to ensure your compliance with governance requirements.

When you create your landing zone, the landing zone and all the organizational units (OUs), accounts, and resources are compliant with the governance rules enforced by your chosen controls. As you and your organization members use the landing zone, changes in this compliance status may occur. Some changes may be accidental, and some may be made intentionally to respond to time-sensitive operational events.

Drift detection assists you in identifying resources that need changes or configuration updates to resolve the drift.

Detecting drift

AWS Control Tower detects drift automatically. To detect drift, the AWSControlTowerAdmin role requires persistent access to your management account so AWS Control Tower can make read-only API calls to AWS Organizations. These API calls show up as AWS CloudTrail events.

Drift is surfaced in the Amazon Simple Notification Service (Amazon SNS) notifications that are aggregated in the audit account. Notifications in each member account send alerts to a local Amazon SNS topic, and to a Lambda function.

For controls that are part of the AWS Security Hub Service-Managed Standard: AWS Control Tower, drift is shown on the Account and Account details pages in the AWS Control Tower console, as well as by means of an Amazon SNS notification.

Member account administrators can (and as a best practice, they should) subscribe to the SNS drift notifications for specific accounts. For example, the aws-controltower-AggregateSecurityNotifications SNS topic provides drift notifications. The AWS Control Tower console indicates to management account administrators when drift has occurred. For more information about SNS topics for drift detection and notification, see Drift prevention and notification (p. 213).

Drift notification de-duplication

If the same type of drift occurs on the same set of resources multiple times, AWS Control Tower sends an SNS notification only for the initial instance of drift. If AWS Control Tower detects that this instance of drift has been remediated, it sends another notification only if drift re-occurs for those identical resources.

Examples: Account drift and SCP drift are handled in the following manner

- If you modify the same managed SCP multiple times, you receive a notification for the first time you modify it.
- If you modify a managed SCP, then remediate drift, then modify it again, you'll receive two notifications.
Types of account drift

- Account moved between OUs
- Account removed from organization

**Note**
When you move an account from one OU to another, the controls from the previous OU are not removed. If you enable any new hook-based control on the destination OU, the old hook-based control is removed from the account, and the new control replaces it. Controls implemented with SCPs and AWS Config rules always must be removed manually when an account changes OUs.

Types of policy drift

- SCP updated
- SCP attached to OU
- SCP detached from OU
- SCP attached to account

For more information, see [Types of Governance Drift (p. 172)].

Resolving drift

Although detection is automatic, the steps to resolve drift must be done through the console.

- Many types of drift can be resolved through the **Landing zone settings** page. You can choose the **Repair** button in the **Versions** section to repair these types of drift.
- If your OU has fewer than 300 accounts, you can repair drift in Account Factory provisioned accounts, or SCP drift, by selecting **Re-register OU** on the **Organization** page or the **OU details** page.
- You may be able to repair account drift, such as **Moved Member Account (p. 173)**, by updating an individual account. For more information, see [Update the account in the console (p. 123)].

**Note**
When you repair your landing zone, the landing zone is upgraded to the latest landing zone version.

Considerations about drift and SCP scans

AWS Control Tower scans your managed SCPs daily to verify that the corresponding controls are applied correctly and that they have not drifted. To retrieve the SCPs and run checks on them, AWS Control Tower calls AWS Organizations on your behalf, using a role in your management account.

If an AWS Control Tower scan discovers drift, you'll receive a notification. AWS Control Tower sends only one notification per drift issue, so if your landing zone already is in a state of drift, you won't receive additional notifications unless a new drift item is found.

AWS Organizations limits how often each of its APIs can be called. This limit is expressed in transactions per second (TPS), and known as the **TPS limit**, **throttling rate**, or **API request rate**. When AWS Control Tower audits your SCPs by calling AWS Organizations, the API calls that AWS Control Tower makes are
counted towards your TPS limit, because AWS Control Tower uses the management account to make the calls.

In rare situations, this limit can be reached when you call the same APIs repeatedly, whether through a third-party solution or a custom script you wrote. For example, if you and AWS Control Tower call the same AWS Organizations APIs at the same moment in time (within 1 second), and the TPS limits are reached, subsequent calls are throttled. That is, these calls return an error such as Rate exceeded.

**If an API request rate is exceeded**

- If AWS Control Tower hits the limit and is throttled, we pause the execution of the audit and resume it at a later time.
- If your workload hits the limit and is throttled, the result can range from slight latency all the way to a fatal error in the workload, depending on how the workload is configured. This edge case is something to be aware of.

**A daily SCP scan consists of**

1. Retrieving all of your OUs.
2. For each registered OU, retrieving all SCPs managed by AWS Control Tower that are attached to the OU. Managed SCPs have identifiers that begin with `aws-guardrails`.
3. For each preventive control enabled on the OU, verifying that the control's policy statement is present in the OU's managed SCPs.

The daily scans consume the TPS for the following AWS Organizations APIs:

<table>
<thead>
<tr>
<th>API Method</th>
<th>Maximum Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>listOrganizationalUnits</td>
<td>8 burst, 5 sustained</td>
</tr>
<tr>
<td>listPoliciesForTarget</td>
<td>8 burst, 5 sustained</td>
</tr>
<tr>
<td>describePolicy</td>
<td>2 TPS</td>
</tr>
</tbody>
</table>

| An OU may have one or more managed SCPs. |

**Types of drift to repair right away**

Most types of drift can be resolved by administrators. A few types of drift must be repaired immediately, including deletion of an organizational unit that the AWS Control Tower landing zone requires. Here are some examples of major drift that you may wish to avoid:

- **Don't delete the Security OU:** The organizational unit originally named **Security** during landing zone setup by AWS Control Tower should not be deleted. If you delete it, you'll see an error message instructing you to repair the landing zone immediately. You won't be able to take any other actions in AWS Control Tower until the repair is complete.

- **Don't delete required roles:** AWS Control Tower checks certain AWS Identity and Access Management (IAM) roles when you log into the console for IAM role drift. If these roles are missing or inaccessible, you'll see an error page instructing you to repair your landing zone. These roles are `AWSControlTowerAdmin`, `AWSControlTowerCloudTrailRole`, `AWSControlTowerStackSetRole`.

For more information about these roles, see [Permissions Required to Use the AWS Control Tower Console](p. 1397).
• Don't delete all Additional OUs: If you delete the organizational unit originally named Sandbox during landing zone setup by AWS Control Tower, your landing zone will be in a state of drift, but you still can use AWS Control Tower. At least one Additional OU is required for AWS Control Tower to operate, but it doesn’t have to be the Sandbox OU.
• Don't remove shared accounts: If you remove shared accounts from Foundational OUs, such as removing the logging account from the Security OU, your landing zone will be in a state of drift and must be repaired before you can continue using the AWS Control Tower console.

Repairable changes to resources

Here's a list of changes to AWS Control Tower resources that are permitted, although they create repairable drift. Results of these permitted operations are viewable in the AWS Control Tower console, although a refresh may be required.

For more information about how to resolve the resulting drift, see Managing Resources Outside of AWS Control Tower.

Changes Permitted Outside the AWS Control Tower Console

• Change the name of a registered OU.
• Change the name of the Security OU.
• Change the name of member accounts in non-Foundational OUs.
• Change the name of AWS Control Tower shared accounts in the Security OU.
• Delete a non-Foundational OU.
• Delete an enrolled account from a non-Foundational OU.
• Change the email address of a shared account in the Security OU.
• Change the email address of a member account in a registered OU.

Note
Moving accounts between OUs is considered drift, and it must be repaired.

Drift and New Account Provisioning

If your landing zone is in a state of drift, the Enroll account feature in AWS Control Tower will not work. In that case, you must provision new accounts through AWS Service Catalog. For instructions, see Provision accounts with AWS Service Catalog Account Factory (p. 122).

In particular, if you've made certain changes to your accounts by means of Service Catalog, such as changing the name of your portfolio, the Enroll account feature will not work.

Types of Governance Drift

Governance drift, also called organizational drift occurs when OUs, SCPs, and member accounts are changed or updated. The types of governance drift that can be detected in AWS Control Tower are as follows:

• Moved Member Account (p. 173)
• Removed Member Account (p. 174)
Another type of drift is *landing zone drift*, which may be found through the management account. Landing zone drift consists of IAM role drift, or any type of organizational drift that specifically affects Foundational OUs and shared accounts.

A special case of landing zone drift is *role drift*, which is detected when a required role is not available. If this type of drift occurs, the console displays a warning page and some instructions on how to restore the role. Your landing zone is unavailable until the role drift is repaired. For more information about drift, see *Don't delete required roles* in the section called *Types of drift to repair right away* (p. 171).

AWS Control Tower does not look for drift regarding other services that work with the management account, including CloudTrail, CloudWatch, IAM Identity Center, AWS CloudFormation, AWS Config, and so forth. No drift detection is available in child accounts, because these accounts are protected by preventive mandatory controls.

However, it does report drift regarding controls that are part of the **AWS Security Hub Service-managed Standard: AWS Control Tower**.

### Moved Member Account

This type of drift occurs on the account rather than the OU. This type of drift can occur when an AWS Control Tower member account, the audit account, or the log archive account is moved from a registered AWS Control Tower OU to any other OU. The following is an example of the Amazon SNS notification when this type of drift is detected.

```json
{
    "Message": "AWS Control Tower has detected that your member account 'account-email@amazon.com (012345678909)' has been moved from organizational unit 'Sandbox (ou-0123-eEXAMPLE)' to 'Security (ou-3210-1EXAMPLE)'. For more information, including steps to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/move-account',
    "ManagementAccountId": "012345678912",
    "OrganizationId": "o-123EXAMPLE",
    "DriftType": "ACCOUNT_MOVED_BETWEEN_OUS",
    "RemediationStep": "Re-register this organizational unit (OU), or if the OU has more than 300 accounts, you must update the provisioned product in Account Factory.",
    "AccountId": "012345678909",
    "SourceId": "012345678909",
    "DestinationId": "ou-3210-1EXAMPLE"
}
```

### Resolutions

When this type of drift occurs for an Account Factory provisioned account in an OU with up to 300 accounts, you can resolve it by:

- Navigating to the **Organization** page in the AWS Control Tower console, selecting the account, and choosing **Update account** at the upper right (fastest option for individual accounts).
• Navigating to the Organization page in the AWS Control Tower console, then choosing Re-register for the OU that contains the account (fastest option for multiple accounts). For more information, see Register an existing organizational unit with AWS Control Tower (p. 190).
• Updating the provisioned product in Account Factory. For more information, see Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog (p. 123).

Note
If you have several individual accounts to update, also see this method for making updates with a script: Provision and update accounts using automation (p. 48).

• When this type of drift occurs in an OU with more than 300 accounts, the drift resolution may depend on which type of account has been moved, as explained in the next paragraphs. For more information, see Update Your Landing Zone (p. 47).
• If an Account Factory provisioned account is moved – In an OU with fewer than 300 accounts, you can resolve the account drift by updating the provisioned product in Account Factory, by re-registering the OU, or by updating your landing zone.

In an OU with more than 300 accounts, you must resolve the drift by making an update to each moved account, either through the AWS Control Tower console or the provisioned product because re-register OU will not perform the update. For more information, see Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog (p. 123).
• If a shared account is moved – You can resolve the drift from moving the audit or log archive account by updating your landing zone. For more information, see Update Your Landing Zone (p. 47).

Deprecated field name
The field name MasterAccountID has been changed to ManagementAccountID to comply with AWS guidelines. The old name is deprecated. Beginning in 2022, scripts that contain the deprecated field name will no longer work.

Removed Member Account
This type of drift can occur when a member account is removed from a registered AWS Control Tower organizational unit. The following example shows the Amazon SNS notification when this type of drift is detected.

```json
{
  "Message" : "AWS Control Tower has detected that the member account 012345678909 has been removed from organization o-123EXAMPLE. For more information, including steps to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/remove-account',
  "ManagementAccountId" : "012345678912",
  "OrganizationId" : "o-123EXAMPLE",
  "DriftType" : "ACCOUNT_REMOVED_FROM_ORGANIZATION",
  "RemediationStep" : "Add account to Organization and update Account Factory provisioned product",
  "AccountId" : "012345678909"
}
```

Resolution
• When this type of drift occurs in a member account, you can resolve the drift by updating the account in the AWS Control Tower console, or in Account Factory. For example, you can add the account to another registered OU from the Account Factory update wizard. For more information, see Update and move account factory accounts with AWS Control Tower or with AWS Service Catalog (p. 123).
• If a shared account is removed from a Foundational OU, you must resolve the drift by repairing your landing zone. Until this drift is resolved, you will not be able to use the AWS Control Tower console.
• For more information about resolving drift for accounts and OUs, see If you manage resources outside of AWS Control Tower (p. 179).

**Note**
In Service Catalog, the Account Factory provisioned product that represents the account is not updated to remove the account. Instead, the provisioned product is displayed as TAINTED and in an error state. To clean up, go to the Service Catalog, choose the provisioned product, and then choose Terminate.

### Unplanned Update to Managed SCP

This type of drift can occur when an SCP for a control is updated in the AWS Organizations console or programmatically using the AWS CLI or one of the AWS SDKs. The following is an example of the Amazon SNS notification when this type of drift is detected.

```json
{
  "Message" : "AWS Control Tower has detected that the managed service control policy 'aws-guardrails-012345 (p-tEXAMPLE)' has been modified. For more information, including steps to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/update-scp',
  "ManagementAccountId" : "012345678912",
  "OrganizationId" : "o-123EXAMPLE",
  "DriftType" : "SCP_UPDATED",
  "RemediationStep" : "Update Control Tower Setup",
  "OrganizationalUnitId" : "ou-0123-1EXAMPLE",
  "PolicyId" : "p-tEXAMPLE"
}
```

### Resolution

When this type of drift occurs in an OU with up to 300 accounts, you can resolve it by:

- Navigating to the **Organization** page in the AWS Control Tower console to re-register the OU (fastest option). For more information, see Register an existing organizational unit with AWS Control Tower (p. 190).
- Updating your landing zone (slower option). For more information, see Update Your Landing Zone (p. 47).

When this type of drift occurs in an OU with more than 300 accounts, resolve it by updating your landing zone. For more information, see Update Your Landing Zone (p. 47).

### SCP Attached to Managed OU

This type of drift can occur when an SCP for a control is attached to any other OU. This occurrence is especially common when you are working on your OUs from outside of the AWS Control Tower console. The following is an example of the Amazon SNS notification when this type of drift is detected.

```json
{
  "Message" : "AWS Control Tower has detected that the managed service control policy 'aws-guardrails-012345 (p-tEXAMPLE)' has been attached to the registered organizational unit 'Sandbox (ou-0123-1EXAMPLE)'. For more information, including steps to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/scp-detached-ou',
  "ManagementAccountId" : "012345678912",
}
```
SCP Detached from Managed OU

This type of drift can occur when an SCP for a control has been detached from an OU that's managed by AWS Control Tower. This occurrence is especially common when you're working from outside of the AWS Control Tower console. The following is an example of the Amazon SNS notification when this type of drift is detected.

```json
{
    "Message" : "AWS Control Tower has detected that the managed service control policy 'aws-guardrails-012345 (p-tEXAMPLE)' has been detached from the registered organizational unit 'Sandbox (ou-0123-1EXAMPLE)'. For more information, including steps to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/scp-detached'",
    "ManagementAccountId" : "012345678912",
    "OrganizationId" : "o-123EXAMPLE",
    "DriftType" : "SCP_DETACHED_FROM_OU",
    "RemediationStep" : "Update Control Tower Setup",
    "OrganizationalUnitId" : "ou-0123-1EXAMPLE",
    "PolicyId" : "p-tEXAMPLE"
}
```

Resolution

When this type of drift occurs in an OU with up to 300 accounts, you can resolve it by:

- Navigating to the OU in the AWS Control Tower console to re-register the OU (fastest option). For more information, see Register an existing organizational unit with AWS Control Tower (p. 190).
- Updating your landing zone (slower option). If the drift is affecting a mandatory control, the update process creates a new service control policy (SCP) and attaches it to the OU to repair the drift. For more information about how to update your landing zone, see Update Your Landing Zone (p. 47).

When this type of drift occurs in an OU with more than 300 accounts, resolve it by updating your landing zone. If the drift is affecting a mandatory control, the update process creates a new service control policy (SCP) and attaches it to the OU to repair the drift. For more information about how to update your landing zone, see Update Your Landing Zone (p. 47).
SCP Attached to Member Account

This type of drift can occur when an SCP for a control is attached to an account in the Organizations console. Guardrails and their SCPs can be enabled on OUs (and thus applied to all of an OU's enrolled accounts) through the AWS Control Tower console. The following is an example of the Amazon SNS notification when this type of drift is detected.

```json
{
    "Message": "AWS Control Tower has detected that the managed service control policy 'aws-guardrails-012345 (p-tEXAMPLE)' has been attached to the member account 'account-email@amazon.com (012345678909)'. For more information, including steps to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/scp-detached-account'",
    "ManagementAccountId": "012345678912",
    "OrganizationId": "o-123EXAMPLE",
    "DriftType": "SCP_ATTACHED_TO_ACCOUNT",
    "RemediationStep": "Re-register this organizational unit (OU)",
    "AccountId": "012345678909",
    "PolicyId": "p-tEXAMPLE"
}
```

Resolution

This type of drift occurs on the account rather than the OU.

When this type of drift occurs for accounts in a Foundational OU, such as the Security OU, the resolution is to update your landing zone. For more information, see [Update Your Landing Zone](p. 47).

When this type of drift occurs in a non-Foundational OU with up to 300 accounts, you can resolve it by:

- Detaching the AWS Control Tower SCP from the account factory account.
- Navigating to the OU in the AWS Control Tower console to re-register the OU (fastest option). For more information, see [Register an existing organizational unit with AWS Control Tower](p. 190).

When this type of drift occurs in an OU with more than 300 accounts, you may attempt to resolve it by updating the account factory configuration for the account. It may not be possible to resolve it successfully. For more information, see [Update Your Landing Zone](p. 47).

Deleted Foundational OU

This type of drift applies only to AWS Control Tower Foundational OUs, such as the Security OU. It can occur if a Foundational OU is deleted outside of the AWS Control Tower console. Foundational OUs cannot be moved without creating this type of drift, because moving an OU is the same as deleting it and then adding it someplace else. When you resolve the drift by updating your landing zone, AWS Control Tower replaces the Foundational OU in the original location. The following example shows an Amazon SNS notification you may receive when this type of drift is detected.

```json
{
    "Message": "AWS Control Tower has detected that the registered organizational unit 'Security (ou-0123-1EXAMPLE)' has been deleted. For more information, including steps to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/delete-ou'",
    "ManagementAccountId": "012345678912",
    "OrganizationId": "o-123EXAMPLE",
    "DriftType": "ORGANIZATIONAL_UNIT_DELETED",
    "RemediationStep": "Delete organizational unit in Control Tower",
    "OrganizationalUnitId": "ou-0123-1EXAMPLE"
}
```
Resolution

Because this drift occurs for Foundational OUs only, the resolution is to update the landing zone. When other types of OUs are deleted, AWS Control Tower is updated automatically.

For more information about resolving drift for accounts and OUs, see If you manage resources outside of AWS Control Tower (p. 179).

Security Hub control drift

This type of drift occurs when a control that's part of the AWS Security Hub Service-Managed Standard: AWS Control Tower reports a state of drift. The AWS Security Hub service itself does not report a state of drift for these controls. Instead, the service sends its findings to AWS Control Tower.

Security Hub control drift also can be detected if AWS Control Tower has not received a status update from Security Hub in more than 24 hours. If those findings are not received as expected, AWS Control Tower verifies that the control is in drift. The following example shows an Amazon SNS notification you may receive when this type of drift is detected.

```
{
"Message": "AWS Control Tower has detected that an AWS Security Hub control was removed in your account example-account@amazon.com <mailto:example-account@amazon.com>. The artifact deployed on the target OU and accounts does not match the expected template and configuration for the control. This mismatch indicates that configuration changes were made outside of AWS Control Tower. For more information, view Security Hub standard",
"MasterAccountId": "123456789XXX",
"ManagementAccountId": "123456789XXX",
"OrganizationId": "o-123EXAMPLE",
"DriftType": "SECURITY_HUB_CONTROL_DISABLED",
"RemediationStep": "To remediate the issue, Re-register the OU, or remove the control and enable it again. If the problem persists, contact AWS support.",
"AccountId": "7876543219XXX",
"ControlId": "PYBETSAGNUZB",
"ControlName": "EBS snapshots should not be publicly restorable",
"ApiControlIdentifier": "arn:aws:controltower:us-east-1::control/PYBETSAGNUZB",
"Region": "us-east-1"
}
```

Resolution

For OUs with fewer than 300 accounts, the resolution is to Re-register the OU, which resets the control to the original state. For any OU, you can remove and re-enable the control through the console or the AWS Control Tower APIs, which also resets the control.

For more information about resolving drift for accounts and OUs, see If you manage resources outside of AWS Control Tower (p. 179).

Trusted access disabled

This type of drift applies to AWS Control Tower landing zones. It occurs when you disable trusted access to AWS Control Tower in AWS Organizations after you set up your AWS Control Tower landing zone.

When trusted access is disabled, AWS Control Tower no longer receives change events from AWS Organizations. AWS Control Tower relies on these change events to stay synchronized with AWS
If you manage resources outside of AWS Control Tower

AWS Control Tower sets up accounts, organizational units, and other resources on your behalf, but you are the owner of these resources. You can change these resources within AWS Control Tower or outside it. The most common place to change resources outside of AWS Control Tower is the AWS Organizations console. This topic describes how to reconcile changes to AWS Control Tower resources when you make the changes outside of AWS Control Tower.

Renaming, deleting, and moving resources outside of the AWS Control Tower console causes the console to become out of sync. Many changes can be reconciled automatically. Certain changes require a repair to your landing zone to update the information that’s displayed in the AWS Control Tower console.

In general, changes that you make outside the AWS Control Tower console to AWS Control Tower resources create a state of repairable drift in your landing zone. For more information about these changes, see Repairable changes to resources (p. 172).

Tasks that require landing zone repair

- Deleting the Security OU (A special case, not to be done lightly.)
- Removing a shared account from the Security OU (Not recommended.)
- Updating, attaching, or detaching an SCP associated with the Security OU.

Changes that are updated automatically by AWS Control Tower

- Changing the email address of an enrolled account
- Renaming an enrolled account
- Creating a new top-level organizational unit (OU)
• Renaming a registered OU
• Deleting a registered OU (Except the Security OU, which requires an update.)
• Deleting an enrolled account (Except a shared account in the Security OU.)

Note
AWS Service Catalog handles changes differently than AWS Control Tower. AWS Service Catalog may create a change in governance posture when it reconciles your changes. For more information about updating a provisioned product, see Updating Provisioned Products in the AWS Service Catalog documentation.

Referring to resources outside of AWS Control Tower

When you create new OUs and accounts outside of AWS Control Tower, they are not governed by AWS Control Tower, even though they may be displayed.

Creating an OU

Organizational Units (OUs) created outside of AWS Control Tower are referred to as Unregistered. They are displayed in the Organization page, but they are not governed by AWS Control Tower controls.

Creating an account

Accounts created outside of AWS Control Tower are referred to as Unenrolled. Enrolled and unenrolled accounts that belong to an OU that's registered with AWS Control Tower are displayed in the Organization page. Accounts that do not belong to a registered OU can be invited by using the AWS Organizations console. This invitation to join does not enroll the account in AWS Control Tower or extend AWS Control Tower governance to the account. To extend governance by enrolling the account, go to the Organization page or the Account detail page in AWS Control Tower and choose Enroll account.

Externally changing AWS Control Tower resource names

You can change the names of your organizational units (OUs) and accounts outside of the AWS Control Tower console, and the console updates automatically to reflect those changes.

Renaming an OU

In AWS Organizations, you can change the name of an OU by using either the AWS Organizations API or the console. When you change an OU name outside of AWS Control Tower, the AWS Control Tower console automatically reflects the name change. However, if you provision your accounts using AWS Service Catalog, you also must repair your landing zone to ensure that AWS Control Tower stays consistent with AWS Organizations. The Repair workflow ensures consistency across services for the Foundational and Additional OUs. You can repair this type of drift from the Landing zone settings page. See the section called "Resolving Drift" in Detect and resolve drift in AWS Control Tower (p. 169).

AWS Control Tower displays the names of OUs on the Organization page in the AWS Control Tower dashboard. You can see when your landing zone repair has succeeded.

Renaming an enrolled account

Each AWS account has a display name that can be changed by the account's root user in the AWS Billing and Cost Management console. When you rename an account that's enrolled in AWS Control Tower, the name change is automatically reflected in AWS Control Tower. For more information about changing an account's name, see Managing an AWS account in the AWS Billing User Guide.
Deleting the Security OU

This type of drift is a special case. If you delete the Security OU, you will see an error message page, prompting you to repair your landing zone. You must repair your landing zone before you can take any other actions in AWS Control Tower.

- You will not be able to perform any actions in the AWS Control Tower console and you will not be able to create any new accounts in AWS Service Catalog until the repair is done.
- You won't be able to view the Landing zone settings page to see the Repair button there.

In this situation, the landing zone repair process creates a new Security OU and moves the two shared accounts into the new Security OU. AWS Control Tower marks the Log Archive and Audit accounts as drifted. The same process repairs the drift in these accounts.

**If you determine that you must delete the Security OU, here's what you need to know:**

Before you can delete the Security OU, you must make sure it contains no accounts. Specifically, you must remove the Log Archive and Audit accounts from the OU. We recommend that you move these accounts to another OU.

**Note**

The action of deleting your Security OU is not to be performed without due consideration. The action could create compliance concerns if logging is suspended temporarily, and because some controls might not be enforced.

For general information about drift, see "Resolving Drift" in Detect and resolve drift in AWS Control Tower (p. 169).

Removing an account from the Security OU

We do not recommend that you remove any of the shared accounts from your organization or move them out of the Security OU. If you have removed a shared account accidentally, you can follow the remediation steps in this section to restore the account.

- **From within the AWS Control Tower console:** To start the remediation process, follow the semi-manual remediation steps. Ensure the user or role you use to access the AWS Control Tower console has permissions to run organizations:InviteAccountToOrganization. If you don't have such permissions, follow the manual remediation steps, which use both the AWS Control Tower console and the AWS Organizations console.

- **Starting from the AWS Organizations console:** This remediation process is a slightly longer, fully manual procedure. When following the manual remediation steps, you'll switch between the AWS Organizations console and the AWS Control Tower console. When working in AWS Organizations, you'll need a user or role with the AWSOrganizationsFullAccess managed policy or equivalent. When working in the AWS Control Tower console, you'll need a user or role with the AWSControlTowerServiceRolePolicy managed policy or equivalent, and permission to run all AWS Control Tower actions (controltower:*).

- If the remediation steps don't restore the account, contact AWS Support.

**The results of removing a shared account through AWS Organizations:**

- The account is no longer protected by AWS Control Tower mandatory controls with service control policies (SCPs). **Result:** The resources created by AWS Control Tower in the account may be modified or deleted.

- The account is no longer under the AWS Organizations management account. **Result:** The administrator of the AWS Organizations management account no longer has visibility into the account's spending.
• The account is no longer guaranteed to be monitored by AWS Config. **Result:** The administrator of the AWS Organizations management account may not be able to detect resource changes.

• The account is no longer in the organization. **Result:** AWS Control Tower updates and repair will fail.

**To restore a shared account using the AWS Control Tower console (semi-manual procedure)**

   You must sign in as an IAM user, user in IAM Identity Center, or role with permissions to run organizations:InviteAccountToOrganization. If you don't have such permissions, use the manual remediation procedure described later in this topic.

2. On the **Landing zone drift detected** page, choose **Re-Invite** to remediate shared account removal by re-inviting the shared account into the organization. An automatically-generated email is sent to the email address for the account.

3. Accept the invitation to bring the shared account back into the organization. Do one of the following:
   - Sign in to the shared account that was removed, then go to [https://console.aws.amazon.com/organizations/home#/invites](https://console.aws.amazon.com/organizations/home#/invites)
   - If you have access to the email message sent when you re-invited the account, sign in to the removed account, then click the link in the message to navigate directly to the account invitation.
   - If the shared account that was removed is not in another organization, sign into the account, open the AWS Organizations console and navigate to **Invitations**.

4. Sign in to the management account again, or reload the AWS Control Tower console if it's already open. You'll see the **Landing zone drift** page. Choose **Repair** to repair the landing zone.

5. Wait for the repair process to complete.

If remediation is successful, the shared account appears in a normal state and compliance.

If the remediation steps don't restore the account, contact AWS Support.

**To restore a shared account using the AWS Control Tower and AWS Organizations consoles (Manual remediation)**

1. Sign in to the AWS Organizations console at [https://console.aws.amazon.com/organizations/](https://console.aws.amazon.com/organizations/). You must sign in as an IAM user, user in IAM Identity Center, or role with the AWSOrganizationsFullAccess managed policy or equivalent.

2. Invite the shared account back to the organization. For information on the requirements, prerequisites, and procedure for inviting an account to AWS Organizations, see **Inviting an AWS account to your organization** in the **AWS Organizations User Guide**.

3. Sign in to the shared account that was removed, then go to [https://console.aws.amazon.com/organizations/home#/invites](https://console.aws.amazon.com/organizations/home#/invites) to accept the invitation.

4. Sign in to the management account again.

5. Sign in to the AWS Control Tower console as a user or role with the AWSControlTowerServiceRolePolicy managed policy or equivalent, and permissions to run all AWS Control Tower actions (controltower:*).

6. You'll see the **Landing zone drift** page with an option to repair the landing zone. Choose **Repair** to repair the landing zone.

7. Wait for the repair process to complete.

If remediation is successful, the shared account appears in a normal state and compliance.

If the remediation steps don't restore the account, contact AWS Support.
External changes that are updated automatically

Changes that you make to your account email addresses are updated by AWS Control Tower automatically, but Account Factory does not update them automatically.

Changing the email address of a governed account

AWS Control Tower retrieves and displays email addresses as required by the console experience. Therefore, shared and other account email addresses are updated and shown consistently in AWS Control Tower after you change them.

**Note**
In AWS Service Catalog, the Account Factory displays the parameters that were specified in the console when you created a provisioned product. However, the original account email address is not updated automatically when the account email address changes. That's because the account is conceptually contained within the provisioned product; it is not the same as the provisioned product. To update this value, you must update the provisioned product, which may cause a change in governance posture.

Applying external AWS Config rules

AWS Control Tower displays the compliance status of all AWS Config rules deployed into organizational units registered with AWS Control Tower, including rules that were activated outside of the AWS Control Tower console.

Deleting AWS Control Tower resources outside AWS Control Tower

You can delete OUs and accounts in AWS Control Tower and you don’t need to take any further action to see the updates. Account Factory is updated automatically when you delete an OU, but not when you delete an account.

**Deleting a registered OU (except the Security OU)**

Within AWS Organizations, you can remove empty organizational units (OUs) by using the API or the console. OUs that contain accounts cannot be deleted.

AWS Control Tower receives a notification from AWS Organizations when an OU is deleted. It updates the OU list in the Account Factory, so that the list of registered OUs remains consistent.

**Note**
In AWS Service Catalog, the Account Factory is updated to remove the deleted OU from the list of available OUs into which you can provision an account.

**Deleting an enrolled account from an OU**

When you delete an enrolled account, AWS Control Tower receives a notification and makes updates, so that the information remains consistent.

**Note**
In AWS Service Catalog, the Account Factory provisioned product that represents the governed account is not updated to delete the account. Instead, the provisioned product is displayed as TAINTED and in an error state. To clean up, go to AWS Service Catalog, choose the provisioned product, and then choose **Terminate**.
Govern organizations and accounts with AWS Control Tower

All organizational units (OUs) and accounts that you create in AWS Control Tower are governed automatically by AWS Control Tower. Also, if you have existing OUs and accounts that were created outside of AWS Control Tower, you can bring them into AWS Control Tower governance.

For existing AWS Organizations and AWS accounts, most customers prefer to enroll groups of accounts by registering the entire organizational unit (OU) that contains the accounts. You also can enroll accounts individually. For more information on enrolling individual accounts, see Enroll an existing AWS account (p. 108).

Terminology

• When you bring an existing organization into AWS Control Tower, it's called registering the organization, or extending governance to the organization.
• When you bring an AWS account into AWS Control Tower, it's called enrolling the account.

View your OUs and accounts

On the AWS Control Tower Organization page, you can view all the OUs in your AWS Organizations, including OUs that are registered with AWS Control Tower and those that are not registered. You can view nested OUs as part of the hierarchy. An easy way to view your organizational units on the Organization page is to select Organizational units only from the dropdown at the upper right.

The Organization page lists all accounts in your organization, regardless of OU or enrollment status in AWS Control Tower. An easy way to view your accounts on the Organization page is to select Accounts only from the dropdown at the upper right. You can view, update, and enroll accounts individually within the OUs, if the accounts meet the prerequisites for enrollment.

If you do not select any filtering, the Organization page displays your accounts and OUs in a hierarchy. It is a central location for monitoring and taking actions on all of your AWS Control Tower resources. For more information about the Organization page, you can view the video walkthrough.

Video Walkthrough

This video (4:01) describes how to work with the Organization page in AWS Control Tower. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Working with the Organization Page in AWS Control Tower.

Topics

• Register an existing organizational unit with AWS Control Tower (p. 190)
• Enroll an existing AWS account (p. 108)

Extend governance to an existing organization

You can add AWS Control Tower governance to an existing organization by setting up a landing zone (LZ) as outlined in the AWS Control Tower User Guide at Getting Started, Step 2.
Here's what to expect when you set up your AWS Control Tower landing zone in an existing organization.

- You can have one landing zone per AWS Organizations organization.
- AWS Control Tower uses the management account from your existing AWS Organizations organization as its management account. No new management account is needed.
- AWS Control Tower sets up two new accounts in a registered OU: an audit account and a logging account.
- Your organization's service limits must allow for the creation of these two additional accounts.
- After you've launched your landing zone or registered an OU, AWS Control Tower controls apply automatically to all enrolled accounts in that OU.
- You can Enroll additional existing AWS accounts into an OU that's governed by AWS Control Tower, so that controls apply to those accounts.
- You can add more OUs in AWS Control Tower and you can Register existing OUs.

To check other prerequisites for registration and enrollment, see Getting Started with AWS Control Tower.

Here's more detail about how AWS Control Tower controls do not apply to your OUs in AWS organizations that don't have AWS Control Tower landing zones set up:

- New accounts created outside of AWS Control Tower Account Factory are not bound by the registered OU's controls.
- New accounts created in OUs that are not registered with AWS Control Tower are not bound by controls, unless you specifically Enroll those accounts into AWS Control Tower. See Enroll an existing AWS account (p. 108) for more information about enrolling accounts.
- Additional existing organizations, existing accounts, and any new OUs or any accounts that you create outside of AWS Control Tower, are not bound by AWS Control Tower controls, unless you separately register the OU or enroll the account.

For more information about how to apply AWS Control Tower to existing OUs and accounts, see Register an existing organizational unit with AWS Control Tower (p. 190).

For an overview of the process of setting up an AWS Control Tower landing zone in your existing organization, see the video in the next section.

**Note**
During set up, AWS Control Tower performs pre-checks to avoid common issues. However, if you are currently using the AWS Landing Zone solution for AWS Organizations, check with your AWS solutions architect before you try to enable AWS Control Tower in your organization to determine if AWS Control Tower may interfere with your current landing zone deployment. Also, see What if the account does not meet the prerequisites? (p. 112) for information about moving accounts from one landing zone to another.

**Video: Enable a Landing Zone in existing AWS Organizations**

This video (7:48), describes how to set up and enable an AWS Control Tower landing zone in existing AWS Organizations structures. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Enable AWS Control Tower for existing organizations
Considerations for IAM Identity Center and existing organizations

- If AWS IAM Identity Center (IAM Identity Center) is already set up, the AWS Control Tower home Region must be the same as the IAM Identity Center Region.
- AWS Control Tower does not delete an existing configuration.
- If IAM Identity Center is already enabled, and if you are using IAM Identity Center Directory, AWS Control Tower adds resources such as permission sets, groups, and so forth, and proceeds as usual.
- If another directory (external, AD, Managed AD) is set up, AWS Control Tower does not change the existing configuration. For more details, see Considerations for AWS IAM Identity Center (IAM Identity Center) customers (p. 17).

Access to other AWS services

After you bring your organization into AWS Control Tower governance, you still have access to any AWS services that are available through AWS Organizations, by means of the AWS Organizations console and APIs. For more information, see Related AWS services (p. 1455).

Nested OUs in AWS Control Tower

This chapter lists the expectations and considerations you'll want to be aware of when working with nested OUs in AWS Control Tower. In most ways, working with nested OUs is the same as working with a flat OU structure. The Register and Re-register features work with nested OUs, except for the changed behaviors that are noted in this chapter.

Video Walkthrough

This video (4:46) describes how to manage nested OU deployments in AWS Control Tower. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Managing Nested OUs in AWS Control Tower.

For guidance regarding best practices for nested OUs and your landing zone, see the blog post Organizing your AWS Control Tower landing zone with nested OUs.

Expand from flat OU structure to nested OU structure

If you created your AWS Control Tower landing zone with a flat OU structure, you can expand it to a nested OU structure.

This process has four main steps:

1. Create your desired nested OU structure in AWS Control Tower.
2. Go to the AWS Organizations console and use their bulk move feature to move the accounts from the source OU (flat) into the destination OU (nested). Here's how:
   a. Go to the OU from which you want to move accounts.
b. Select all the accounts in the OU.

c. Choose **Move**.

**Note**
This step must be done in the in AWS Organizations console because AWS Control Tower doesn’t have a **Move** feature.

3. Go to the nested OU in AWS Control Tower and **Register** or **Re-register** it. All of the accounts in the nested OU will be enrolled.

- If you created the OU in AWS Control Tower, **Re-register** the OU.
- If you created the OU in AWS Organizations, **Register** the OU for the first time.

4. After your accounts are moved and enrolled, delete the empty top-level OU, either from the AWS Organizations console or from the AWS Control Tower console.

## Nested OU registration pre-checks

To support successful registration of your nested OUs and their member accounts, AWS Control Tower performs a series of pre-checks. These same prechecks are performed when registering any top-level OU or nested OU. For more information, see [Common causes of failure during registration or re-registration](p. 192).

- If all pre-checks pass, AWS Control Tower begins registering your OU, automatically.
- If any pre-checks fail, AWS Control Tower stops the registration process and provides you with a list of items that must be fixed before you can register your OU.

## Nested OUs and roles

AWS Control Tower deploys the `AWSControlTowerExecution` role to accounts under the target OU, and to accounts in all OUs nested under the target OU, even when your intention is to register the target OU only. This role gives any user of the management account **Administrator** permissions on any account that has the `AWSControlTowerExecution` role. The role can be used to perform actions that normally would be disallowed by AWS Control Tower controls.

You can delete this role from unenrolled accounts that you don’t plan to enroll. If you delete this role, you cannot enroll the account with AWS Control Tower, or register the immediate parent OUs, unless you restore the role to the account. To delete the `AWSControlTowerExecution` role from an account, you must be signed in under the `AWSControlTowerExecution` role, because no other IAM principals are allowed to delete roles managed by AWS Control Tower.

For information about how to restrict role access, see [Optional conditions for your role trust relationships](p. 88).

## What happens during registration and re-registration of nested OUs and accounts

When you register or re-register a nested OU, AWS Control Tower enrolls all unenrolled accounts of the target OU, and it updates all enrolled accounts. Here’s what to expect.

**AWS Control Tower performs the following tasks**

- Adds the `AWSControlTowerExecution` role to all unenrolled accounts under this OU, and to all unenrolled accounts in its nested OUs.
• Enrolls member accounts that are not enrolled.
• Re-enrolls enrolled member accounts.
• Creates an IAM Identity Center login for newly enrolled member accounts.
• Updates existing enrolled member accounts to reflect your landing zone changes.
• Updates controls that are configured for this OU and its member accounts.

Considerations for nested OU registration

• You cannot register an OU under the core OU (Security OU).
• Nested OUs must be registered separately.
• You cannot register an OU unless its parent OU is registered.
• You cannot register an OU unless all OUs higher in the tree have been registered successfully at some time (some may have been deleted).
• You can register an OU that is under a drifted higher OU, but the drift is not repaired by that action.

Nested OU limitations

• OUs may be nested a maximum of 5 levels deep under the root.
• Nested OUs under the target OU must be registered or re-registered separately.
• If the target OU is at Level 2 or below in the hierarchy, that is, if it is not a top-level OU, preventive controls enabled on higher OUs are enforced on this OU and all OUs below it, automatically.
• OU registration failures do not propagate up the hierarchy tree. You can see details about the states of nested OUs on the parent's OU details page.
• OU registration failures do not propagate down the hierarchy tree.
• AWS Control Tower does not modify your VPC settings for any new or existing accounts.

Nested OUs and compliance

From the AWS Control Tower console, you can view OUs and accounts that are non-compliant in the Organization page, so you can understand compliance at a larger scale.

Considerations about compliance for nested OUs and accounts

• An OU's compliance is not determined based on the compliance of the OUs nested under it.
• A control's compliance status is computed over all OUs on which the control is enabled, including nested OUs. See AWS Control Tower compliance status for controls, OUs, and accounts (p. 212).
• An OU is shown as noncompliant only if it has accounts that are noncompliant, regardless of where the OU sits in the OU hierarchy.
• If a nested OU is noncompliant, its parent OU is not automatically considered to be noncompliant.
• On the OU detail or Account detail page, you can view a list of noncompliant resources that may be causing your OUs or accounts to show a non-compliant status.

Nested OUs and drift

In certain situations, drift can prevent the registration of nested OUs.
Expectations for drift and nested OUs

- You can enable controls on OUs with drifted parents, but not on drifted OUs directly.
- You are allowed to enable detective controls under a drifted OU, as long as it’s not a top-level drifted OU.
- Mandatory controls are enabled on top-level OUs only. Mandatory controls are skipped when you register a nested OU.
- One mandatory control protects AWS Config resources; therefore, that control must be in a non-drifted state to register nested OUs. If drifted, AWS Control Tower blocks registration of nested OUs.
- If the top-level OU is in drift, the control that protects AWS Config resources may be in drift. In this situation, AWS Control Tower blocks any action that requires creation or update of AWS Config resources, including application of detective controls.

Nested OUs and controls

When you enable a control on a registered OU, preventive and detective controls have different behaviors. For nested OUs, proactive controls behave similarly to detective controls.

Preventive controls

- Preventive controls are enforced on nested OUs.
- Mandatory preventive controls are enforced on all accounts under the OU and its nested OUs.
- Preventive controls affect all accounts and OUs nested under the target OU, even if those accounts and OUs are not registered.

Detective and proactive controls

- Nested OUs do not inherit detective or proactive controls automatically; these must be enabled separately.
- Detective and proactive controls are deployed only to registered accounts in your landing zone's operating Regions.

Enabled control states and inheritance

You can view inherited controls for each OU, on the OU details page.

Tip
You can make use of control inheritance to help stay within an OU's SCP quota. For example, you can enable a control at the top-level OU of an OU hierarchy, instead of enabling directly for a nested OU.

Inherited status

- The status Inherited indicates that the control is enabled by inheritance only, and it has not been applied directly to the OU.
- The status Enabled means the control is enforced on this OU, regardless of its state on other OUs.
- The status Failed means the control is not enforced on this OU, regardless of its state on other OUs.

Note
The status Inherited indicates that the control was applied to an OU higher in the tree, and it is enforced on this OU, but it was not added directly to this OU.

If your landing zone is not the current version
Each row in the Enabled controls table represents one enabled control on one, individual OU.
Nested OUs and the root

The root is not an OU, and it cannot be registered or re-registered. You also can’t create accounts directly in the root. The root cannot be noncompliant or have a lifecycle state, such as registered or in drift.

However, the root is the top-level container for all accounts and OUs. In the context of nested OUs, it is the node under which all other OUs are nested.

Register an existing organizational unit with AWS Control Tower

An efficient way to bring multiple, existing AWS accounts into AWS Control Tower is to extend governance by AWS Control Tower to an entire organizational unit (OU).

To enable AWS Control Tower governance over an existing OU that was created with AWS Organizations, and its accounts, register the OU with your AWS Control Tower landing zone. You can register OUs that contain up to 300 accounts. If an OU contains more than 300 accounts, you cannot register it in AWS Control Tower.

When you register an OU, its member accounts are enrolled into the AWS Control Tower landing zone. They are governed by the controls that apply to their OU.

**Note**

If you don’t already have an AWS Control Tower landing zone, start by setting up a landing zone, either in a new organization created by AWS Control Tower, or in an existing AWS Organizations organization. For more details about how to set up a landing zone, see Getting started with AWS Control Tower (p. 16).

What happens to my accounts when I register my OU?

AWS Control Tower requires permission to establish trusted access between AWS CloudFormation and AWS Organizations on your behalf, so that AWS CloudFormation can deploy your stack to the accounts in your organization automatically.

- The AWSControlTowerExecution role is added to all accounts with status Not enrolled.
- Mandatory controls are enabled by default to your OU and all its accounts when you register your OU.

Partial enrollment of accounts after an OU is registered

It’s possible to register an OU successfully, yet certain accounts may remain unenrolled. If so, these accounts do not meet some of the prerequisites for enrollment. If an account enrollment as part of the Register OU process does not succeed, the account status on the accounts page shows Enrollment failed. You may also see account information on your OU page such as 4 of 5, in the accounts field.

For example, if you see 4 of 5, it means that your OU has 5 accounts in total, and 4 of them enrolled successfully, but one account failed to enroll during the Register OU process. You can choose Re-Register OU to bring accounts into enrollment, after you make sure the accounts meet the enrollment prerequisites.

IAM user prerequisites for registering an OU

Your AWS Identity and Access Management (IAM) identity (user or role) or IAM Identity Center user identity must be included on the appropriate Account Factory portfolio when you perform the Register OU operation, even if you already have Admin permissions. Otherwise, the creation of the provisioned
Register an existing OU

In the AWS Control Tower console, on the Organization page, you can view all of your organization's OUs and accounts in a hierarchy, including OUs that are registered with AWS Control Tower, and those that are not registered.

In general, unregistered OUs were created in AWS Organizations, and they are not governed by any other landing zone. You can register existing OUs that contain up to 300 accounts. If an OU contains more than 300 accounts, you cannot register it in AWS Control Tower.

To register an existing OU

2. In the left-pane navigation menu, choose Organization.
3. On the Organization page, select the radio button next to the OU you want to register, then select Register organizational unit from the Actions dropdown menu at the upper right, or alternatively, select the name of the OU so you can view the OU details page for that OU.
4. On the OU details page, at the upper right you can select Register OU from the Actions dropdown menu.

The registration process takes a minimum of 10 minutes to extend governance to the OU, and up to 2 additional minutes for each additional account.

Results of registering an existing OU

After you register an existing OU, the AWSControlTowerExecution role allows AWS Control Tower to extend governance to its individual accounts. Guardrails are enforced, and information about account activities is reported to your audit and logging accounts.

Other results include the following:

- AWSControlTowerExecution allows auditing by the AWS Control Tower audit account.
- AWSControlTowerExecution helps you configure your organization's logging, so that all the logs for every account are sent to the logging account.
- AWSControlTowerExecution ensures that your selected AWS Control Tower controls apply automatically to every individual account in your OUs, as well as to every new account you create in AWS Control Tower.

For a registered OU, you can provide compliance and security reports based on the auditing and logging features embodied by AWS Control Tower controls. Your security and compliance teams can verify that all requirements are met, and that no organizational drift has occurred. For more information about drift, see Detect and resolve drift in AWS Control Tower (p. 169).

Note

One unusual situation can occur when AWS Control Tower displays OUs and their accounts. If you have created an account in a registered OU and then you subsequently move that enrolled account into another OU that's not registered, particularly if you use AWS Organizations to
Create a new OU

To create a new OU in AWS Control Tower

1. Navigate to the Organization page.
2. Select Create organizational unit from the Create resources dropdown menu in the upper right.
3. Specify a name in the OU name field.
4. In the Parent OU dropdown, you can see the hierarchy of registered OUs. Select a parent OU for the new OU you’re creating.
5. Choose Add.

Tip
To add a nested OU in fewer steps, select the name of the parent OU shown in the table on the Organization page, view the OU page for that parent OU, and then choose Add an OU from the Actions dropdown menu in the upper right. The new OU is created as a nested OU under your selected OU, automatically.

Note
If your landing zone is not up to date, you will see a flat list instead of a hierarchy in the dropdown menu. Even if your landing zone includes nested OUs, you will not see L5 OU’s in the dropdown, because you cannot create a new OU beneath a L5 OU. For more information about nested OUs in AWS Control Tower, see Nested OUs in AWS Control Tower (p. 186).

Common causes of failure during registration or re-registration

If registration (or re-registration) of an OU or any of its member accounts fails, you can download a file containing a detailed report that shows which pre-checks did not pass. You can complete the download by choosing the Download button, which appears at the upper right of the registration area.

This section lists the types of errors you may receive if pre-checks fail, and how to correct the errors.

In general, when you register or re-register an OU, all accounts within that OU are enrolled in AWS Control Tower. However, it is possible that some accounts may fail to enroll, even if the OU as a whole is registered successfully. In these cases, you must resolve the pre-check failure related to the account and then try re-enrolling that account or OU.

Landing Zone error

- Landing zone not ready
  Repair your current landing zone, or update it to the latest version.

OU errors

- Exceeds maximum number of SCPs
  You may be over the limit for service control policies (SCPs) per OU, or you may have reached another quota. A limit of 5 SCPs per OU applies to all OUs in your AWS Control Tower landing zone. If you have more SCPs than the quota allows, you must delete or combine the SCPs.
Common causes of failure during registration or re-registration

• **Conflicting SCPs**

Existing SCPs may be applied to the OU or account, which prevent AWS Control Tower from enrolling the account. Check the applied SCPs for any policy that may prevent AWS Control Tower from working. Be sure to check the SCPs that are inherited from OUs higher in the hierarchy.

• **Exceeds stack set quota**

The stack set quota may have been exceeded. If you have more instances than the quota allows, you must delete some stack instances. For more information, see [AWS CloudFormation quotas](awscontroltower.amazonaws.com) in the AWS CloudFormation User Guide.

• **Exceeds account limit**

AWS Control Tower limits each OU to 300 accounts during registration.

**Account errors**

• **Pre-checks prevented on accounts**

An existing SCP on the OU prevents AWS Control Tower from conducting pre-checks on your OU member accounts. To resolve this pre-check failure, update or remove the SCP from the OU.

• **Email address error**

The email address you specified for the account does not conform to the naming standards. Here is the regular expression (regex) that specifies which characters are allowed: `[A-Z0-9a-z._%+-]+@[A-Za-z0-9.-]+[.]+[A-Za-z]+`

• **Config recorder or delivery channel enabled**

The account may have an existing AWS Config configuration recorder or delivery channel. These must be deleted or modified through the AWS CLI in all AWS Regions where the AWS Control Tower management account has governed resources, before you can enroll an account.

• **STS disabled**

AWS Security Token Service (AWS STS) may be disabled in the account. AWS STS endpoints must be activated in the accounts for all Regions supported by AWS Control Tower.

• **IAM Identity Center conflict**

The AWS Control Tower home Region is not the same as the AWS IAM Identity Center (IAM Identity Center) Region. If IAM Identity Center is already set up, the AWS Control Tower home region must be the same as the IAM Identity Center Region.

• **Conflicting SNS topic**

The account has an Amazon Simple Notification Service (Amazon SNS) topic name that AWS Control Tower needs to use. AWS Control Tower creates resources (such as SNS topics) with specific names. If these names are already taken, AWS Control Tower setup fails. This situation could occur if you are reusing an account previously enrolled in AWS Control Tower.

• **Suspended account detected**

This account has been suspended. It cannot be enrolled into AWS Control Tower. Remove the account from this OU, and try again.

• **IAM user not in portfolio**

Add the AWS Identity and Access Management (IAM) user to the Service Catalog portfolio before registering your OU. This error pertains to the management account only.

• **Account does not meet prerequisites**
The account doesn’t meet prerequisites for account enrollment. For example, the account may be missing roles and permissions required to enroll it in AWS Control Tower. Instructions for adding a role are available in *Manually add the required IAM role to an existing AWS account and enroll it (p. 113).*

As a reminder, AWS CloudTrail is auto-enabled on all of your AWS accounts when you enroll them in AWS Control Tower. If CloudTrail is enabled on an account previous to enrollment, you could experience double-billing unless you deactivate CloudTrail before you begin the enrollment process.

**Update organizations**

The quickest way to update an organizational unit (OU) or to update multiple accounts within an OU is to **Re-register** the OU.

**When to update AWS Control Tower OUs and accounts**

When you perform a landing zone update, you must update your enrolled accounts to apply new controls to those accounts.

- You can perform an update to all accounts under an OU using the **Re-Register** option.
- If you have more than one registered OU in your landing zone, re-register all of your OUs to update all of your accounts.
- To update a single account, you can update from the AWS Control Tower console, or you can select the *Update provisioned product* option in AWS Service Catalog. See *Update the account in the console (p. 123).*

**Update multiple accounts in the same OU**

**To update multiple accounts in one OU, with one action**

2. In the left-pane navigation menu, choose **Organization**.
3. On the **Organization** page, choose any OU to view the **OU details** page.
4. Under **Actions** in the upper right, select **Re-Register OU**.

Repeat these steps for each OU in your AWS Control Tower organization, if you need to update all of your accounts and OUs.

Alternatively, you can select any account that shows a status of **Update available** and then choose **Update account** for as many accounts as needed.

**What happens during re-registration**

**When you re-register an OU:**

- The **State** field indicates whether the account currently is enrolled with AWS Control Tower (**Enrolled**), whether the account has never been enrolled (**Not enrolled**), or whether enrollment failed previously (**Enrollment failed**).
When you re-register the OU, the AWSControlTowerExecution role is added to all accounts with status **Not enrolled** or **Enrollment failed**.

AWS Control Tower creates a single sign-on (IAM Identity Center) login for those new enrolled accounts.

**Enrolled** accounts are re-enrolled into AWS Control Tower.

Drift on any preventive controls applied to the OU is fixed, because the SCPs are returned to their default definitions.

All accounts are updated to reflect the latest landing zone changes.

For more information, see *Enroll an existing AWS account (p. 108)*.

**Tip**

When you re-register an OU, or when you’re updating your landing zone version and multiple member accounts, you may see a failure message mentioning the **StackSet-AWSControlTowerExecutionRole**. This StackSet in the management account can fail because the **AWSControlTowerExecution** IAM role already exists in all enrolled member accounts. This error message is expected behavior, and it can be disregarded.

### Update a single account

You can update individual AWS Control Tower accounts in the AWS Control Tower console, or in the Service Catalog console.

To update a single account in the AWS Control Tower console, see *Update the account in the console (p. 123)*.

To update a single account in AWS Service Catalog

1. Go to AWS Service Catalog.
2. In the left-pane navigation menu, choose **Provisioned products**.
3. On the **Provisioned products** page, select the radio button next to the provisioned product you want to update.
4. In the upper right, choose the **Actions** dropdown to **Update**.

To learn more about updating in AWS Service Catalog, see *Update the provisioned product (p. 123)* and *Updating products* in the *Service Catalog Administrator Guide*. 
About controls in AWS Control Tower

A control is a high-level rule that provides ongoing governance for your overall AWS environment. It's expressed in plain language. AWS Control Tower implements preventive, detective, and proactive controls that help you govern your resources and monitor compliance across groups of AWS accounts.

A control applies to an entire organizational unit (OU), and every AWS account within the OU is affected by the control. Therefore, when users perform work in any AWS account in your landing zone, they're always subject to the controls that are governing their account's OU.

Note
We are transitioning our terminology to align better with industry usage and with other AWS services. During this time, you may see the previous term, guardrail, as well as the new term, control, in our documentation, console, blogs, and videos. These terms are synonymous for our purposes.

The purpose of controls
Controls assist you to express your policy intentions. For example, if you enable the detective control Detect Whether Public Read Access to Amazon S3 Buckets is Allowed on an OU, you can determine whether an entity (such as a user) would be permitted to have read access over the internet to any Amazon S3 buckets, for any accounts under that OU.

Control behavior and guidance

Controls are categorized according to their behavior and their guidance.

The behavior of each control is one of preventive, detective, or proactive. Control guidance refers to the recommended practice for how to apply each control to your OUs. The guidance of a control is independent of whether its behavior is preventive, detective, or proactive.

Control behavior

- **Preventive** – A preventive control ensures that your accounts maintain compliance, because it disallows actions that lead to policy violations. The status of a preventive control is either enforced or not enabled. Preventive controls are supported in all AWS Regions.

- **Detective** – A detective control detects noncompliance of resources within your accounts, such as policy violations, and provides alerts through the dashboard. The status of a detective control is either clear, in violation, or not enabled. Detective controls apply only in those AWS Regions supported by AWS Control Tower.

- **Proactive** – A proactive control scans your resources before they are provisioned, and makes sure that the resources are compliant with that control. Resources that are not compliant will not be provisioned. Proactive controls are implemented by means of AWS CloudFormation hooks, and they apply to resources that would be provisioned by AWS CloudFormation. The status of a proactive control is PASS, FAIL, or SKIP. For more information about AWS CloudFormation hooks, see Characteristics of hooks in the AWS CloudFormation documentation.

Implementation of control behavior
• The preventive controls are implemented using Service Control Policies (SCPs), which are part of AWS Organizations.

• The detective controls are implemented using AWS Config rules.

• The proactive controls are implemented using AWS CloudFormation hooks.

• Certain mandatory controls are implemented by means of a single SCP that performs multiple actions, rather than as unique SCPs. Therefore, the same SCP is shown in the control reference, under each mandatory control to which that SCP applies.

• The integrated, detective Security Hub controls are implemented using AWS Config rules, similarly to all Security Hub controls. These controls are owned by the Service-Managed Standard: AWS Control Tower, which is part of Security Hub.

Control guidance

AWS Control Tower provides three categories of guidance: mandatory, strongly recommended, and elective controls.

• Mandatory controls are always enforced in your landing zone. You cannot turn them off for any OU.

• Strongly recommended controls are designed to enforce some common best practices for well-architected, multi-account environments. These controls apply at the OU level, for all accounts in that OU.

• Elective controls enable you to track or lock down actions that are commonly restricted in an AWS enterprise environment. These controls apply at the OU level, for all accounts in that OU.

Defaults: When you create a new landing zone, AWS Control Tower enables all mandatory controls by default and applies them to your top-level OUs. When you extend governance to an OU, AWS Control Tower applies mandatory controls to the OU by default. Strongly recommended and elective controls are not enabled by default.

Considerations for controls and OUs

When working with controls and OUs, consider the following properties:

Controls, landing zones, and OUs

• After you create your landing zone, all resources in your landing zone are subject to controls. For example, certain controls apply to Amazon S3 buckets.

• OUs created through AWS Control Tower have mandatory controls applied to them automatically. Optional controls are applied at the discretion of administrators.

• OUs created outside of an AWS Control Tower landing zone (such as, unregistered OUs created in AWS Organizations) are displayed in the AWS Control Tower console, but AWS Control Tower controls do not apply to those OUs, unless they become registered OUs.

• Regarding nested OUs, preventive controls enabled on any OUs higher in the tree will apply to unregistered OUs in that tree.

• When you enable controls on an organizational unit (OU) that is registered with AWS Control Tower, preventive controls apply to all member accounts under the OU, enrolled and unenrolled. Detective controls apply to enrolled accounts only.

For more information about how controls are applied to nested OUs, in AWS Control Tower, see Nested OUs and controls (p. 189).
Exception to controls for the management account

The root user and any administrators in the management account can perform work that controls would otherwise deny. This exception is intentional. It prevents the management account from entering into an unusable state. All actions taken within the management account continue to be tracked in the logs contained within the log archive account, for purposes of accountability and auditing.

Considerations for controls and accounts

When working with controls and accounts, consider the following properties:

Controls and accounts

- Accounts created through the Account Factory in AWS Control Tower inherit the controls of the parent OU, and the associated resources are created.
- Accounts created outside of an AWS Control Tower landing zone do not inherit AWS Control Tower controls. These are called unenrolled accounts.
- Accounts created outside of AWS Control Tower won't inherit controls in AWS Control Tower until you enroll them. However, these unenrolled accounts are displayed in AWS Control Tower.

Accounts inherit controls from an OU upon enrollment in that OU.

- An OU can contain enrolled or unenrolled member accounts.
- Controls do not apply to an unenrolled account unless it becomes a member account of a registered AWS Control Tower OU. In that case, preventive controls for the OU will apply to the unenrolled account. Detective controls will not apply.
- When you enable optional controls, AWS Control Tower creates and manages certain additional AWS resources in your accounts. Do not modify or delete resources created by AWS Control Tower. Doing so could result in the controls entering an unknown state. For more information, see The AWS Control Tower controls library (p. 215).
- When you move an account from one OU to another, the controls from the previous OU are not removed. If you enable any new hook-based control on the destination OU, the old hook-based control is removed from the account, and the new control replaces it. Controls implemented with SCPs and AWS Config rules always must be removed manually when an account changes OUs.

View control details

To view details about an individual control in the console, select the name of the control from the table on the Controls page.

To view more details about an individual control in the AWS Control Tower User Guide, see Tables of control metadata (p. 1501). For each control, the API controlIdentifier for each Region is available, along with the framework and objective. Certain additional information is available only in the console, as described in the next sections.

In each Control details page of the console, you can find the following details for each control:

- **Name** – The name of the control.
- **Control objective** – The pre-defined objective that this control helps you enforce. See the List of control objectives (p. 199).
- **Service** – The AWS service to which this control applies.
• **Control owner** – The AWS service that owns and maintains this control.
• **Behavior** – A control's behavior is set to preventive, detective, or proactive.
• **Implementation** – The underlying implementation method for this control, such as SCP, AWS Config managed rule, or AWS CloudFormation hook.
• **Resource** – The AWS resource that is monitored or affected by this control.
• **Framework** – The industry-standard compliance framework that this control helps to enforce, for example, NIST 800-53 Rev 5.
• **Control ID** – A unique identifier assigned to each control. This identifier is part of a classification system for the controls.
• **API controlIdentifier** – This identifier is needed when calling the AWS Control Tower APIs.
• **Guidance** – The guidance is either mandatory, strongly recommended, or elective.
• **Severity** – The relative risk associated with any violation of this control.
• **Release date** – The date the control became available.

The status of the Region deny control is shown as a separate entry.

Other information may appear on the Control details page, including these:

• **Description** – A brief description of the control and its function.
• **Remediation message** – Suggestions for what to change if your AWS CloudFormation hook control returns a FAIL status.
• **Remediation samples** – Examples showing configurations that can return a PASS or FAIL result for your AWS CloudFormation hook control.
• **Usage considerations** – Additional information about how to apply this control or about the resources it can affect.
• **The Gherkin artifact** – The Gherkin is a readable specification for the AWS CloudFormation hook controls, showing requirements for tests that cause PASS, FAIL, or SKIP results to be returned.

**To view a control artifact**

Each control is implemented by one or more artifacts. These artifacts can include a baseline AWS CloudFormation template, a service control policy (SCP) to prevent account-level configuration changes or activity that may create configuration drift, and AWS Config Rules to detect account-level policy violations.

To view a control's artifact, select the Artifact tab to view the Service control policy (SCP), AWS Config rule, or AWS CloudFormation policy template on the Control details page.

**List of control objectives**

Each control enforces one of these objectives. Sometimes controls must be applied in a group so that the control objective is enforced. Information about related controls is viewable in the AWS Control Tower console, on the Control details page.

**Control objectives**

For more information about controls and their associated control objectives, see Tables of control metadata (p. 1501).

- **CO.1** Establish logging and monitoring
- **CO.2** Encrypt data at rest
• CO.3 Encrypt data in transit
• CO.4 Protect data integrity
• CO.5 Enforce least privilege
• CO.6 Limit network access
• CO.7 Optimize costs
• CO.8 Improve resiliency
• CO.9 Improve availability
• CO.10 Protect configurations
• CO.11 Prepare for incident response
• CO.12 Manage vulnerabilities
• CO.13 Manage secrets
• CO.14 Prepare for disaster recovery
• CO.15 Use strong authentication

Resource identifiers for APIs and controls

Each control in AWS Control Tower has a unique identifier for use with the control APIs. A different identifier is given for each Region in which AWS Control Tower operates. The identifier for each control is shown in the API controlIdentifier field, on the Control details page in the AWS Control Tower console, and in the Tables of control metadata (p. 1501).

Note
This identifier is distinct from the ControlID field, which is a classification system for controls.

View the control identifiers for all controls

To view the tables of control metadata, including the controlIdentifier ARN for each control and Region, see Tables of control metadata (p. 1501). The tables also include the identifiers for Security Hub controls that are part of the AWS Security Hub Service-Managed Standard: AWS Control Tower.

View control identifiers in the console

To view the control identifiers and other details about AWS Control Tower controls in the console, navigate to the Control details page in the AWS Control Tower console. You can find the identifier in the API controlIdentifier field.

Example forms of Identifiers
When you look in the AWS Control Tower console, here are examples of identifiers you may see.

• Security Hub example API controlIdentifier: arn:aws:controltower:us-east-1::control/OOTDCUSIKIZZ
• Legacy control example API controlIdentifier: arn:aws:controltower:us-east-1::control/AWS-GR_LOG_GROUP_POLICY
• Proactive control example API controlIdentifier: arn:aws:controltower:us-east-1::control/EHSOKSSMVFWF

Older controls (legacy controls) include the name of the control in the ARN, but newer controls have a UUID, and that is expected.
Old example: arn:aws:controltower:us-east-1::control/AWS-GR_CLOUDTRAIL_CHANGE_PROHIBITED
New example: arn:aws:controltower:us-east-1::control/WTDSMKDKDNLE
The following list contains the API controlIdentifier designations of the (legacy) **Strongly recommended** and **Elective**, **preventive** and **detective**, controls that are owned by AWS Control Tower, including the elective **Data residency** controls. Mandatory controls cannot be deactivated by the control APIs.

Each item in the list that follows serves as a link, which provides more information about these individual (legacy) controls that are owned by AWS Control Tower, as given in the [AWS Control Tower controls library](p. 215).

### Designations for legacy Elective controls

- `arn:aws:controltower:REGION::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED`
- `arn:aws:controltower:REGION::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED`
- `arn:aws:controltower:REGION::control/AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED`
- `arn:aws:controltower:REGION::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY`
- `arn:aws:controltower:REGION::control/AWS-GR_IAM_USER_MFA_ENABLED`
- `arn:aws:controltower:REGION::control/AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS`
- `arn:aws:controltower:REGION::control/AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION`
- `arn:aws:controltower:REGION::control/AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA`
- `arn:aws:controltower:REGION::control/AWS-GR_S3_VERSIONING_ENABLED`

### Designations for legacy Data residency controls (elective)

- `arn:aws:controltower:REGION::control/AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED`
- `arn:aws:controltower:REGION::control/AWS-GR_AUTOSCALING_LAUNCH_CONFIG_PUBLIC_IP_DISABLED`
- `arn:aws:controltower:REGION::control/AWS-GR_DISALLOW_CROSS_REGION_NETWORKING`
- `arn:aws:controltower:REGION::control/AWS-GR_DISALLOW_VPC_INTERNET_ACCESS`
- `arn:aws:controltower:REGION::control/AWS-GR_DISALLOW_VPN_CONNECTIONS`
- `arn:aws:controltower:REGION::control/AWS-GR_DMS_REPLICATION_NOT_PUBLIC`
- `arn:aws:controltower:REGION::control/AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK`
- `arn:aws:controltower:REGION::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP`
- `arn:aws:controltower:REGION::control/AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS`
- `arn:aws:controltower:REGION::control/AWS-GR_ELASTICSEARCH_IN_VPC_ONLY`
- `arn:aws:controltower:REGION::control/AWS-GR_EMR_MASTER_NO_PUBLIC_IP`
- `arn:aws:controltower:REGION::control/AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED`
- `arn:aws:controltower:REGION::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW`
- `arn:aws:controltower:REGION::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK`
- `arn:aws:controltower:REGION::control/AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC`
- `arn:aws:controltower:REGION::control/AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS`
- `arn:aws:controltower:REGION::control/AWS-GR_SSM_DOCUMENT_NOT_PUBLIC`

### Designations for legacy Strongly recommended controls

- `arn:aws:controltower:REGION::control/AWS-GR_ENCRYPTED_VOLUMES`
Controls that cannot be changed with the AWS Control Tower APIs

The following controls cannot be activated or deactivated by means of the AWS Control Tower APIs. Except for the Region deny control, all of these are mandatory controls. In general, mandatory controls cannot be deactivated. The Region deny control must be changed in the console.

- AWS-GR_REGION_DENY
- AWS-GR_AUDIT_BUCKET_DELETION_PROHIBITED
- AWS-GR_AUDIT_BUCKET_PUBLIC_READ_PROHIBITED
- AWS-GR_AUDIT_BUCKET_PUBLIC_WRITE_PROHIBITED
- AWS-GR_CLOUDTRAIL_CHANGE_PROHIBITED
- AWS-GR_CLOUDTRAIL_CLOUDWATCH_LOGS_ENABLED
- AWS-GR_CLOUDTRAIL_ENABLED
- AWS-GR_CLOUDTRAIL_VALIDATION_ENABLED
- AWS-GR_CLOUDWATCH_EVENTS_CHANGE_PROHIBITED
- AWS-GR_CONFIG_AGGREGATION_AUTHORIZATION_POLICY
- AWS-GR_CONFIG_AGGREGATION_CHANGE_PROHIBITED
- AWS-GR_CONFIG_CHANGE_PROHIBITED
- AWS-GR_CONFIG_ENABLED
- AWS-GR_CONFIG_RULE_CHANGE_PROHIBITED
- AWS-GR_CT_AUDIT_BUCKET_ENCRYPTION_CHANGES_PROHIBITED
- AWS-GR_CT_AUDIT_BUCKET_LIFECYCLE_CONFIGURATION_CHANGES_PROHIBITED
- AWS-GR_CT_AUDIT_BUCKET_LOGGING_CONFIGURATION_CHANGES_PROHIBITED
- AWS-GR_CT_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED
- AWS-GR_IAM_ROLE_CHANGE_PROHIBITED
- AWS-GR_LAMBDA_CHANGE_PROHIBITED
- AWS-GR_LOG_GROUP_POLICY
- AWS-GR_SNS_CHANGE_PROHIBITED
Find identifiers for OUs

For more information about how to find the resource identifier for an OU and its resources, see Resource types defined by AWS Organizations.

To learn more about how to get information from an OU, see the AWS Organizations API Reference.

Note
The control State and status information is available in the console only. It is not available from the public API. To view the status of a control, navigate to the Control details page in the AWS Control Tower console.

Control API examples

Each control in AWS Control Tower has a unique identifier for use with the control APIs. The identifier for each control is shown in the API controlIdentifier field, on the Control details page in the AWS Control Tower console. This identifier is distinct from the ControlID field, which is a classification system for controls.

Note
When you invoke EnableControl on an account or OU, the operationIdentifier value is returned by means of ListEnabledControls or GetEnabledControl even if the enable operation fails. In the AWS Control Tower console, you can determine whether the EnableControl operation was successful, by verifying that the control is enabled on the account or OU. Programatically, you can track the status of the EnableControl operation with the GetControlOperation API command, by passing it the value of operationIdentifier as shown in an example that follows.

EnableControl

For more information about this API operation, see EnableControl.

Example input for EnableControl:

This example shows how to specify the control you wish to enable, and activate that control for the target OU that you identify.

```
{
  controlIdentifier: "arn:aws:controltower:us-west-2::control/AWS-GR_AUTOSCALING_LAUNCH_CONFIG_PUBLIC_IP_DISABLED",
  targetIdentifier: "arn:aws:organizations::123456789123:ou/o-kg8aXXXXXX/ou-prlj-a5kXXXXX"
}
```

Example output for EnableControl:

As an example of how to use this output parameter, you can pass the operationIdentifier parameter as an input to the GetControlOperation API, to track the status of your EnableControl task.

```
{
  "operationIdentifier":"e2bXXXXX-6cab-XXXX-bde7-XX0c6fXXXXXX"
}
```
Example CLI command:

```
aws controltower enable-control \
--control-identifier arn:aws:controltower:us-west-2::control/AWS-
GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED \
--target-identifier arn:aws:organizations::123456789123:ou/o-qnilXXXXXX/ou-vwxu-qqlXXXXX \
--region us-west-2
```

**DisableControl**

For more information about this API operation, see [DisableControl](#).

**Example input for DisableControl:**

```
{
  controlIdentifier: "arn:aws:controltower:us-west-2::control/AWS-
  GR_AUTOSCALING_LAUNCH_CONFIG_PUBLIC_IP_DISABLED",
  targetIdentifier: "arn:aws:organizations::123456789123:ou/o-kg8aXXXXXX/ou-prlj-
a5kxxxxx"
}
```

**Example output for DisableControl:**

```
{
  "operationIdentifier":"e2bxxxxx-8xai-xxxx-bde7-xx0c6fxxxxxx"
}
```

**GetControlOperation**

For more information about this API operation, see [GetControlOperation](#).

**Example input for GetControlOperation:**

When you give an `operationIdentifier` as input, you receive a status message as output.

```
{
  operationIdentifier: "e2bxxxxx-6cab-xxxx-bde7-xx0c6fxxxxxx"
}
```

**Example output for GetControlOperation:**

```
{
  "ControlOperationStatus":{
    "OperationType": "ENABLE_CONTROL",
    "StartTime": "2022-02-02T20:52:08.034Z",
    "Status": "IN_PROGRESS"
  }
}
```

**Example output for GetControlOperation:**

```
{
  "ControlOperationStatus": {
    "EndTime": "2022-04-28T19:36:31Z",
    "OperationType": "DISABLE_CONTROL",
    "Status": "SUCCESS"
  }
}
```
GetEnabledControl

For more information about this API operation, see GetEnabledControl.

Example for GetEnabledControl

```bash
aws controltower get-enabled-control --enabled-control-identifier arn:aws:controltower:us-east-1:123456789012:enabledcontrol/49DVF3KP34ANNC57
```

ListEnabledControls

For more information about this API operation, see ListEnabledControls.

Example input for ListEnabledControls:

This example shows how to specify the target OU as input, so you can receive a list of controls as output.
targetIdentifier: "arn:aws:organizations::123456789123:ou/o-kg8aXXXXXX/ou-prlj-a5kXXXXX",
nextToken: "bde7-XX0c6fxXXXXX",
maxResults: 2
}

Example output for ListEnabledControls:

{
  "enabledControls": [
    {
      "controlIdentifier": "arn:aws:controltower:us-west-2::control/AWS-GR_AUTOSCALING_LAUNCH_CONFIG_PUBLIC_IP_DISABLED"
    },
    {
      "controlIdentifier": "arn:aws:controltower:us-west-2::control/AWS-GR_RESTRICT_ROOT_USER"
    }
  ],
  "nextToken": "e2bXXXXX-6cab-XXXX"
}

This example shows a larger set of returned values for ListEnabledControls.

aws controltower list-enabled-controls --target-identifier arn:aws:organizations::072569612342:ou/o-yy6713pfv2/ou-slt4-fl6mi3bd --max-items 3
{
  "enabledControls": [
    {
      "arn": "arn:aws:controltower:us-west-2::enabledcontrol/SOME_ENABLED_CONTROL",
      "controlIdentifier": "arn:aws:controltower:us-west-2::control/SOME_CONTROL",
      "targetIdentifier": "arn:aws:organizations::072569612342:ou/o-yy6713pfv2/ou-slt4-fl6mi3bd",
      "statusSummary": {
        "status": "SUCCEEDED",
        "lastOperationIdentifier": "12e51344-a73a-439a-8477-fb3cd7f8b410"
      },
      "driftStatusSummary": {
        "driftStatus": "NOT_CHECKING"
      }
    },
    {
      "arn": "arn:aws:controltower:us-west-2::enabledcontrol/OTHER_ENABLED_CONTROL",
      "controlIdentifier": "arn:aws:controltower:us-west-2::control/OTHER_CONTROL",
      "targetIdentifier": "arn:aws:organizations::072569612342:ou/o-yy6713pfv2/ou-slt4-fl6mi3bd",
      "statusSummary": {
        "status": "FAILED",
        "lastOperationIdentifier": "12e51344-a73a-439a-8477-fb3cd7f8b410"
      },
      "driftStatusSummary": {
        "driftStatus": "UNKNOWN"
      }
    },
    {
      "arn": "arn:aws:controltower:us-west-2::enabledcontrol/ANOTHER_ENABLED_CONTROL",
      "controlIdentifier": "arn:aws:controltower:us-west-2::control/ANOTHER_CONTROL",
      "targetIdentifier": "arn:aws:organizations::072569612342:ou/o-yy6713pfv2/ou-slt4-fl6mi3bd",
      "statusSummary": {
        "status": "SUCCEEDED",
        "lastOperationIdentifier": "12e51344-a73a-439a-8477-fb3cd7f8b410"
      }
    }
  ]
}
Enable controls with AWS CloudFormation

You can enable controls with AWS CloudFormation, either through the AWS CloudFormation console, or through the AWS CLI. This section gives an example of each type.

Each control in AWS Control Tower has a unique identifier for use with the control APIs. The identifier for each control is shown in the API controlIdentifier field, on the Control details page in the AWS Control Tower console. This identifier is distinct from the ControlID field, which is a classification system for controls.

Create the stack through AWS CloudFormation

You can use AWS CloudFormation to help you enable AWS Control Tower controls. Here's an example template.

```yaml
Resources:
  TestControl:
    Type: AWS::ControlTower::EnabledControl
    Properties:
      ControlIdentifier: arn:aws:controltower:us-west-2::control/AWS-GR_RESTRICT_ROOT_USER
      TargetIdentifier: arn:aws:organizations::123456789012:ou/o-ybfpt9XXXl/ou-XXXcnVuY2F0ZV9hbW91bnQiOiAyfQ==
```

To create your stack through the AWS CloudFormation console, edit the template to contain the control and target of your choice, then save the template with the file name template.yaml. Follow the AWS CloudFormation wizard. When the wizard asks for a template file, enter the file you saved as template.yaml. For more information, see Creating a stack on the Amazon CloudFormation console.

**Note**
The limit for EnableControl and DisableControl updates in AWS Control Tower is 10 concurrent operations.

Create the stack through AWS CloudFormation and the AWS CLI

Here's an example of creating the stack with the CLI.

```bash
aws cloudformation create-stack --region us-west-2 --stack-name testControlTower --template-body "$(cat << TEMPLATE
Resources:
  TestControl:
    Type: AWS::ControlTower::EnabledControl
    Properties:
      ControlIdentifier: arn:aws:controltower:us-west-2::control/AWS-GR_RESTRICT_ROOT_USER
      TargetIdentifier: arn:aws:organizations::123456789012:ou/o-ybfpt9XXXl/ou-XXXcnVuY2F0ZV9hbW91bnQiOiAyfQ==
 TEMPLATE)"
```

You can also save the example template as a template.yaml file, then upload your file to an Amazon S3 bucket. Later, you can provide the URL for the bucket with the --template-url flag.
Enable controls on an OU

Mandatory and inherited controls are enabled automatically according to an OU's configuration. Optional controls can be enabled manually on your OUs, from the console, or by means of the control APIs. The following procedure describes the steps for enabling controls on an OU, from the console.

Important
When you enable optional controls, AWS Control Tower creates and manages AWS resources in your accounts. Do not modify or delete resources created by AWS Control Tower. Doing so could result in the controls entering an unknown state.

To enable controls in an OU, from the console
2. From the left navigation, choose Controls.
3. Choose a control that you want to enable; for example, Control: Detect Whether Encryption is Enabled for Amazon EBS Volumes Attached to Amazon EC2 Instances. This choice opens the control's details page.
4. From Organizational units enabled, choose Enable control on OU.
5. A new page is displayed that lists the names of your OUs. Identify the OU on which you want to enable this control.
6. Choose Enable control on OU.
7. Your control is now enabled. It may take several minutes for the change to complete. When it does, you'll see that this control is applied to the OU you selected.

Note
You can enable preventive and detective controls concurrently.

To deactivate controls for an OU, from the console
2. From the left navigation, choose Controls.
3. Choose a control that you want to deactivate; for example, Control: Detect Whether Encryption is Enabled for Amazon EBS Volumes Attached to Amazon EC2 Instances. This choice opens the control's details page.
4. From the Organizational units enabled tab, select the radio button next to the OU from which you want to remove the control.
5. Choose Disable control at the upper right.
6. Your control is now deactivated. It may take several minutes for the change to complete. When it does, you'll see that this control is no longer applied to the OU you selected.

Concurrent deployment for optional controls

When applying optional controls, you can deploy more than one control at a time. For example, you can enable Detect Whether MFA for the Root User is Enabled and Detect Whether Public Write Access to Amazon S3 Buckets is Allowed, at the same time.

You can apply and remove multiple optional controls without waiting for individual control operations to complete, and up to 25 control operations are queued. The only restricted times are when AWS Control Tower is in the process of landing zone setup, or while extending governance to a new organization.

Functionality available in the console and with APIs

- Apply and remove different detective controls on the same OU, concurrently.
- Apply and remove different detective controls on different OUs, concurrently.
- Apply and remove the same detective control on multiple OUs, concurrently.
- Apply and remove different preventive controls on the same OU, concurrently.
- Apply and remove different preventive controls on different OUs, concurrently.
- Apply and remove the same preventive control on multiple OUs, concurrently.
- Apply and remove different proactive controls on the same OU, concurrently.
- Apply and remove different proactive controls on different OUs, concurrently.
- Apply and remove the same proactive control on multiple OUs, concurrently.
- You can apply and remove preventive, detective, and proactive controls, concurrently.

When you apply preventive controls to nested OUs, the preventive controls affect all accounts and OUs nested under the target OU, even if those accounts and OUs are not registered with AWS Control Tower. Preventive controls are implemented using Service Control Policies (SCPs), which are part of AWS Organizations. Detective controls are implemented using AWS Config rules. Proactive controls are implemented using CloudFormation hooks.

Controls remain in effect for the entire OU, as you create new accounts or make changes to your existing accounts, and AWS Control Tower provides a summary report of how each account conforms to your enabled controls. For a full list of available controls, see The AWS Control Tower controls library (p. 215).

Controls and compliance

Within AWS Control Tower, compliance refers to the state of a resource, when it is evaluated with respect to a deployed detective control, or a drift detection rule. Compliance in AWS Control Tower is related to drift — usually, a non-compliant resource is in a state of drift. AWS Control Tower controls embody rules of compliance. They help you identify compliant and non-compliant resources by helping identify drift.

When AWS Control Tower evaluates the compliance of resources, it reports the compliance results at the OU, account, and control levels. This section describes compliance status in detail, for controls, OUs, and accounts.
How can administrators review compliance?

Compliance reporting is intended to let cloud administrators know when the resources associated with the accounts in their organization are compliant with established policies. When the resources are in compliance, builders can provision new AWS accounts quickly in a few clicks.

When we talk about compliance in AWS Control Tower, we do not intend the same meaning as compliance with governmental regulations, such as data privacy or health information standards. However, AWS Control Tower can assist your organization to comply with many governmental regulations, sometimes referred to as frameworks.

- For more information about how AWS Control Tower helps you maintain compliance with governmental regulations and industry standards, see Compliance Validation.
- For more information about how you can verify AWS Control Tower resource compliance during AWS CloudFormation stack creation, see this blog post, How AWS Control Tower users can proactively verify compliance in AWS CloudFormationstacks.

For ongoing governance, administrators can enable pre-configured controls—clearly defined rules for security, operations, and compliance. These controls can:

- Prevent deployment of resources that don’t conform to policies (by means of preventive controls, implemented with SCPs, or by means of proactive controls, implemented with AWS CloudFormation hooks).
- Continuously monitor deployed resources for nonconformance (by means of detective controls, implemented with AWS Config rules).

Examples of compliance rules (controls) in AWS Control Tower:

- Detect Whether Public Write Access to Amazon S3 Buckets is Allowed (p. 1366)
- Detect Whether Unrestricted Incoming TCP Traffic is Allowed (p. 1362)

Examples of governmental compliance regulations (frameworks):

- The U.S. Health Insurance Portability and Accountability Act of 1996 (HIPAA)
- The European Union's General Data Protection Regulation of 2016 (GDPR)

How can administrators review compliance?

Compliance with detective controls is determined according to data retrieved from the AWS Config aggregator in the AWS Control Tower Audit account. You can review compliance status in the AWS Control Tower console, by subscribing to SNS topics that send email messages to the Audit account, or both.

Detective control status

To view the compliance status of detective controls in the AWS Control Tower console, select Controls in the left navigation, choose the control name from the controls table, and then scroll to the Accounts section on that control details page. Accounts may show a control compliance status of Unknown if any detective controls are misconfigured. For example, status Unknown often can appear due to account drift, such as Moved account drift. The Unknown status also can appear as a result of SCP drift.

Note

AWS Control Tower displays the compliance status of all AWS Config rules deployed into organizational units registered with AWS Control Tower, including rules that were activated outside of the AWS Control Tower console. To view the compliance status of all your Config rules, navigate to the Account details page in the AWS Control Tower console. You will see a
How can administrators review compliance?

A list showing the compliance status of controls managed by AWS Control Tower and Config rules deployed outside of AWS Control Tower. You can identify any non-compliant AWS Config rule.

Preventive control status

The compliance status of preventive controls on an OU may be viewed on the OU detail page, by scrolling to the Enabled controls section. If any preventive controls are misconfigured for an OU, the State field for that OU may show the state of Registration failed, in the Details section near the top of the page. Preventive control misconfiguration is caused most often by SCP drift, which can occur if the control's SCP is modified or detached from the OU by means of the AWS Organizations console.

Proactive control status

The control compliance status also can be viewed on other pages:

- On the AWS Control Tower Dashboard page, by scrolling to the Controls section near the bottom of the page.
- On the Control details page, which you can view by selecting the name of a control on another page.

Note

The State of a control, as viewed in the AWS Control Tower console, reflects only the enabled or de-activated state of the control for a specific OU. This field does not reflect any information about the framework compliance status or the drift status of the landing zone environment. The control State and Status information is available in the console only. It is not available from the public API. To view the control status, navigate to the Control details page in the AWS Control Tower console.

Nested OUs and compliance

When an OU shows a status of Noncompliant, it means that one of the accounts directly under the OU contains noncompliant resources. The compliance status of an OU is not influenced by the compliance status of nested OUs under the OU, or the compliance status of any accounts that are not directly under the OU.

Other resources

If an account has any non-compliant resources, that account may be shown with Noncompliant status on the OU or Account page in the AWS Control Tower console. Details about the specific resources that have caused the non-compliant status are shown on the Account details page.

If an account shows Compliant status, that means it has no resources that are non-compliant; therefore, no resource details are shown on the Account details page, only an empty table.

Receive compliance status updates

To receive updates about compliance, you can subscribe to SNS topics that send notifications when resource compliance status changes. See Compliance notifications by SNS in the audit account (p. 214), later in this chapter.

For more information on how AWS Control Tower collects information about resources, see the AWS Config Aggregator Documentation.

Drift changes the compliance status for OU and account resources

Drifted resources may be shown with status Unknown in the Compliance status field of the AWS Control Tower console. The Unknown state indicates that AWS Control Tower cannot determine the compliance status of the resource, because drift is present. Drift is not necessarily a detective control compliance violation. For more information about drift, see Detect and resolve drift in AWS Control Tower (p. 169).

In another case of this type of drift, resources may be shown as compliant when they are not. If you delete an AWS Config rule, or if you turn off the Config recorder, compliance status may
be reflected inaccurately in the console, because compliance no longer can be evaluated. For example, if you turn off the Config recorder, the last evaluated status continues to appear in the console. Similarly, if you delete an AWS Config rule, the resources covered by that rule always show to be compliant. In this situation, your environment could have some non-compliant resources that are not reported. Avoid deleting or turning off your AWS Config resources.

AWS Control Tower compliance status for controls, OUs, and accounts

Compliance is reported in the AWS Control Tower dashboard for controls, accounts, and OUs. This section lists the possible categories of compliance and non-compliance in AWS Control Tower, assuming that controls are enabled for an account or an OU.

- **For a control:** A compliance status of **Enforced**, **Clear**, or **In violation** is possible, as long as the control is enabled on your OUs and the member accounts in the OUs. You can view this control status on individual control detail pages.

- **For an account or OU:** A compliance status of **Compliant**, **Noncompliant**, or **Unknown** is possible. The compliance status refers to the status of the resources associated with a single account, or the status of all accounts in an OU that has multiple controls enabled on it. The account or OU compliance status can be found on the account or OU detail pages.

**Note**
The **State** of a control, as viewed in the AWS Control Tower console, reflects only the enabled or de-activated state of the control for a specific OU. This field does not reflect any information about the framework compliance status or the drift status of the landing zone environment. The control **State** and **Status** information is available in the console only. It is not available from the public API. To view the control status, navigate to the **Control details** page in the AWS Control Tower console.

The following list gives more information about compliance status as reported specifically for controls.

- **Enforced** – Maximum level of protection. Operations that would break this compliance rule are simply not allowed.
  - **Reported for:** Preventive controls (SCPs)
  - **Applies to:** Any preventive control that's enabled on any of multiple accounts that are members of an OU. Controls are enabled at the OU level.

- **Clear** – Compliance rules are properly in place. No violations have been detected.
  - **Reported for:** Detective controls (AWS Config Rules)
  - **What is checked:** Any detective control that's enabled on any of multiple accounts that are members of an OU. Controls are enabled at the OU level.

- **In violation** – Denotes that resources are actively breaching a control.
  - **Reported for:** Detective controls (AWS Config Rules)
  - **What is checked:** Any detective control that's enabled on any of multiple accounts that are members of an OU. Controls are enabled at the OU level.

The following list gives more information about compliance status as reported specifically for OUs and their member accounts.

- **Compliant** – Compliance rules are properly in place. No violations have been detected for any resources. Controls are applied at the OU level, for all enrolled accounts in the OU, and their resources.
  - **Reported for:** Detective controls (AWS Config Rules)
• **What it checks:**
  • Any individual detective control that's applied to the member accounts in an OU
  • Multiple detective controls that are applied to the member accounts in an OU
• **Noncompliant** – Compliance rules are in place. However, non-compliant resources have been detected in one or more member accounts in the OU.
• **Reported for:** Detective controls (AWS Config Rules)

**What it checks:**
• Any individual detective control that's applied to the member accounts in an OU
• Multiple detective controls that are applied to the member accounts in an OU

The following status can be reported for any account, control, or OU.

**Unknown** – A compliance rule is broken or compliance cannot be guaranteed.

• **Reported for:**
  • Detective controls (AWS Config Rules)
  • Preventive controls (SCPs)
• **What it checks:**
  • Any detective control that's enabled on any accounts that are members of an OU. Controls are enabled at the OU level.
  • Any preventive control that's enabled on any accounts that are members of an OU. Controls are enabled at the OU level.
  • Basically anything with a compliance status (account, control, resource, or OU).

**Drift prevention and notification**

You can enable certain controls and subscribe to certain SNS notifications that help you maintain compliance in AWS Control Tower.

**Drift monitoring protection**

AWS Control Tower provides passive and active methods of drift monitoring protection for preventive controls.

• **Passive protection:** AWS Organizations monitors and logs preventive control (SCP) drift.
• **Active protection:** The AWS Control Tower drift monitoring service (p. 170) actively scans the preventive control SCPs, on a regular basis.

AWS Control Tower notifies you by means of SNS messaging, if drift is detected.

**Drift prevention**

Some controls prevent modification of compliance reporting mechanisms.

• **Disallow Changes to AWS Config Rules Set Up by AWS Control Tower (p. 225)** (Mandatory, preventive control)
• **Disallow Deletion of AWS Config Aggregation Authorizations Created by AWS Control Tower (p. 219)** (Mandatory, preventive control)
• **Disallow Changes to Tags Created by AWS Control Tower for AWS Config Resources (p. 223)** (Mandatory, preventive control)
• **Disallow Configuration Changes to AWS Config** *(Mandatory, preventive control)*

In contrast to preventive controls, detective controls notify you of resources that violate the associated AWS Config rule.

**To receive SNS notifications about drift and control compliance**

For information about how to receive appropriate drift and control compliance notifications by Amazon SNS, see *Compliance notifications by SNS in the audit account* *(p. 214)*.

**Publishers and subscribers for SNS topics**

**The aws-controltower-AllConfigNotifications topic:**

- The `AWS::Config::DeliveryChannel` resource is configured to send notifications about configuration changes to this topic.
- The possible types of notifications that AWS Config can send are defined in the Amazon SNS Topic section of the AWS Config documentation.
- The `AWS::CloudTrail::Trail` resource is configured to send notifications of log file delivery to this topic.
- You may subscribe to this topic.

**The aws-controltower-SecurityNotifications topic:**

- The `AWS::Events::Rule` resource is configured to send notifications about AWS Config Rule compliance changes (one of the SNS notification types) to this topic.
- The `aws-controltower-NotificationForwarder` Lambda function is subscribed to this topic, and it forwards the SNS notifications to the `aws-controltower-AggregateSecurityNotifications` topic.

**The aws-controltower-AggregateSecurityNotifications topic:**

- This topic receives notifications from `aws-controltower-SecurityNotifications`, forwarded by the Lambda function.
- It also receives drift notifications in the home Region.
- When AWS Control Tower creates the topic, a subscription is added for the audit account email address, and you must confirm the subscription.

**Note**

The endpoint, such as an email address, must confirm each subscription, SNS doesn't send messages to an endpoint until the subscription is confirmed.

**Compliance notifications by SNS in the audit account**

To receive compliance change notifications in email sent to your audit account, subscribe to this Amazon SNS topic:


When subscribing, substitute your actual AWS Control Tower home Region and audit account information into the topic name shown. You can subscribe to SNS topics that receive notifications about each supported AWS Region in which you run AWS Control Tower.
SNS topics and notifications you can receive

- The aws-controltower-AllConfigNotifications topic:
  It receives notifications from AWS Config regarding compliance, noncompliance, and change. It also receives notification from AWS CloudTrail on log file delivery.
- The aws-controltower-SecurityNotifications topic:
  One of these topics exists for each supported AWS Region. It receives compliance, noncompliance, and change notifications from AWS Config in that Region. It forwards all incoming notifications to aws-controltower-AggregateSecurityNotifications.
- The aws-controltower-AggregateSecurityNotifications topic:
  This topic exists in each supported AWS Region. It receives compliance change notifications from the region-specific aws-controltower-SecurityNotifications topics. Additionally, in the home Region, it also receives drift notifications.

Other considerations about SNS topics:

- All of these topics exist and receive notifications in the Audit account.
- By default, the Audit account email address is subscribed to the aws-controltower-AggregateSecurityNotifications SNS topic.
- SNS topics in AWS Control Tower are extremely noisy, by design. For example, AWS Config sends a notification every time AWS Config discovers a new resource.
- Administrators who wish to filter out specific types of notifications from an SNS topic can create an AWS Lambda function and subscribe it to the SNS topic. Alternatively, you can set up an EventBridge rule to filter notifications, as described in this support article, How can I be notified when an AWS resource is non-compliant using AWS Config?
- AWS Config notifications contain a JSON object.
- AWS Control Tower drift notifications appear in plain text.

The AWS Control Tower controls library

The following sections include an individual reference entry for each of the controls available in AWS Control Tower. The controls are grouped into sections according to common characteristics. Each control reference entry includes the details, artifacts, additional information, and considerations to keep in mind when enabling a specific control on a OU in your landing zone.

For summary tables of control information, see Tables of control metadata (p. 1501). Additional detail is available in the AWS Control Tower console.

Topics

- Mandatory controls (p. 216)
- Proactive controls (p. 230)
- Security Hub standard (p. 1341)
- Controls that enhance data residency protection (p. 1343)
- Optional controls (p. 1360)

Note
The four mandatory controls with "Sid": "GRCLOUDTRAILENABLED" are identical by design. The sample code is correct.
Mandatory controls

Mandatory controls are owned by AWS Control Tower, and they apply to every OU on your landing zone. These controls are applied by default when you set up your landing zone, and they can't be deactivated. Following, you'll find a reference for each of the mandatory controls available in AWS Control Tower.

Topics

- Disallow Changes to Encryption Configuration for AWS Control Tower Created Amazon S3 Buckets in Log Archive (p. 216)
- Disallow Changes to Logging Configuration for AWS Control Tower Created Amazon S3 Buckets in Log Archive (p. 217)
- Disallow Changes to Bucket Policy for AWS Control Tower Created Amazon S3 Buckets in Log Archive (p. 217)
- Disallow Changes to Lifecycle Configuration for AWS Control Tower Created Amazon S3 Buckets in Log Archive (p. 218)
- Disallow Changes to Amazon CloudWatch Logs Log Groups set up by AWS Control Tower (p. 218)
- Disallow Deletion of AWS Config Aggregation Authorizations Created by AWS Control Tower (p. 219)
- Disallow Deletion of Log Archive (p. 219)
- Detect Public Read Access Setting for Log Archive (p. 220)
- Detect Public Write Access Setting for Log Archive (p. 220)
- Disallow Configuration Changes to CloudTrail (p. 221)
- Integrate CloudTrail Events with Amazon CloudWatch Logs (p. 221)
- Enable CloudTrail in All Available Regions (p. 222)
- Enable Integrity Validation for CloudTrail Log File (p. 222)
- Disallow Changes to Amazon CloudWatch Set Up by AWS Control Tower (p. 223)
- Disallow Changes to Tags Created by AWS Control Tower for AWS Config Resources (p. 223)
- Disallow Configuration Changes to AWS Config (p. 224)
- Enable AWS Config in All Available Regions (p. 224)
- Disallow Changes to AWS Config Rules Set Up by AWS Control Tower (p. 225)
- Disallow Changes to AWS IAM Roles Set Up by AWS Control Tower and AWS CloudFormation (p. 226)
- Disallow Changes to AWS Lambda Functions Set Up by AWS Control Tower (p. 228)
- Disallow Changes to Amazon SNS Set Up by AWS Control Tower (p. 228)
- Disallow Changes to Amazon SNS Subscriptions Set Up by AWS Control Tower (p. 229)
- Detect whether shared accounts under the Security organizational unit have AWS CloudTrail or CloudTrail Lake enabled (p. 229)

Note

The four mandatory controls with "Sid": "GRCLOUDTRAILENABLED" are identical by design. The sample code is correct.

Disallow Changes to Encryption Configuration for AWS Control Tower Created Amazon S3 Buckets in Log Archive

This control prevents changes to encryption for the Amazon S3 buckets that AWS Control Tower creates in the log archive account. This is a preventive control with mandatory guidance. By default, this control is enabled on the Security OU. It cannot be enabled on additional OUs.

The artifact for this control is the following service control policy (SCP).
Mandatory controls

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCTAUDITBUCKETENCRYPTIONCHANGESPROHIBITED",
            "Effect": "Deny",
            "Action": ["s3:PutEncryptionConfiguration"],
            "Resource": ["arn:aws:s3:::aws-controltower*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Disallow Changes to Logging Configuration for AWS Control Tower Created Amazon S3 Buckets in Log Archive

This control prevents changes to logging configuration for the Amazon S3 buckets that AWS Control Tower creates in the log archive account. This is a preventive control with mandatory guidance. By default, this control is enabled on the Security OU. It cannot be enabled on additional OUs.

The artifact for this control is the following SCP.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCTAUDITBUCKETLOGGINGCONFIGURATIONCHANGESPROHIBITED",
            "Effect": "Deny",
            "Action": ["s3:PutBucketLogging"],
            "Resource": ["arn:aws:s3:::aws-controltower*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Disallow Changes to Bucket Policy for AWS Control Tower Created Amazon S3 Buckets in Log Archive

This control prevents changes to bucket policy for the Amazon S3 buckets that AWS Control Tower creates in the log archive account. This is a preventive control with mandatory guidance. By default, this control is enabled on the Security OU. It cannot be enabled on additional OUs.

The artifact for this control is the following SCP.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCTAUDITBUCKETPOLICYPROHIBITED",
            "Effect": "Deny",
            "Action": ["s3:PutBucketPolicy"],
            "Resource": ["arn:aws:s3:::aws-controltower*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```
Disallow Changes to Lifecycle Configuration for AWS Control Tower Created Amazon S3 Buckets in Log Archive

This control prevents lifecycle configuration changes for the Amazon S3 buckets that AWS Control Tower creates in the log archive account. This is a preventive control with mandatory guidance. By default, this control is enabled on the Security OU. It cannot be enabled on additional OUs.

The artifact for this control is the following SCP.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRCTAUDITBUCKETLIFECYCLECONFIGURATIONCHANGESPROHIBITED",
      "Effect": "Deny",
      "Action": [
        "s3:PutLifecycleConfiguration"
      ],
      "Resource": ["arn:aws:s3:::aws-controltower*"],
      "Condition": {
        " ArnNotLike": {
          "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
  ]
}
```

Disallow Changes to Amazon CloudWatch Logs Log Groups set up by AWS Control Tower

This control prevents changes to the retention policy for Amazon CloudWatch Logs log groups that AWS Control Tower created in the log archive account when you set up your landing zone. It also prevents modifying the log retention policy in customer accounts. This is a preventive control with mandatory guidance. By default, this control is enabled on all OUs.

The artifact for this control is the following SCP.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRCTAUDITBUCKETLIFECYCLECONFIGURATIONCHANGESPROHIBITED",
      "Effect": "Deny",
      "Action": [
        "s3:PutLifecycleConfiguration"
      ],
      "Resource": ["arn:aws:s3:::aws-controltower*"],
      "Condition": {
        " ArnNotLike": {
          "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
  ]
}
```
Disallow Deletion of AWS Config Aggregation Authorizations Created by AWS Control Tower

This control prevents deletion of AWS Config aggregation authorizations that AWS Control Tower created in the audit account when you set up your landing zone. This is a preventive control with mandatory guidance. By default, this control is enabled on all OUs.

The artifact for this control is the following SCP.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRCONFIGAGGREGATIONAUTHORIZATIONPOLICY",
         "Effect": "Deny",
         "Action": [
            "config:DeleteAggregationAuthorization"
         ],
         "Resource": [
            "arn:aws:config:*::*:aggregation-authorization*"
         ],
         "Condition": {
            "ArnNotLike": {
               "aws:PrincipalArn": "arn:aws:iam::*:role/AWSControlTowerExecution"
            },
            "StringLike": {
               "aws:ResourceTag/aws-control-tower": "managed-by-control-tower"
            }
         }
      }
   }
}
```

Disallow Deletion of Log Archive

This control prevents deletion of Amazon S3 buckets created by AWS Control Tower in the log archive account. This is a preventive control with mandatory guidance. By default, this control is enabled on the Security OU.
The artifact for this control is the following SCP.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRAUDITBUCKETDELETIONPROHIBITED",
            "Effect": "Deny",
            "Action": [
                "s3:DeleteBucket"
            ],
            "Resource": [
                "arn:aws:s3:::aws-controltower*"
            ],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

**Detect Public Read Access Setting for Log Archive**

This control detects whether public read access is enabled to the Amazon S3 buckets in the log archive shared account. This control does not change the status of the account. This is a detective control with mandatory guidance. By default, this control is enabled on the **Security OU**.

The artifact for this control is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check that your S3 buckets do not allow public access.

Parameters:
ConfigRuleName:
  Type: 'String'
  Description: 'Name for the Config rule'

Resources:
CheckForS3PublicRead:
  Type: AWS::Config::ConfigRule
  Properties:
    ConfigRuleName: !Sub ${ConfigRuleName}
    Description: Checks that your S3 buckets do not allow public read access. If an S3 bucket policy or bucket ACL allows public read access, the bucket is noncompliant.
    Source:
      Owner: AWS
      SourceIdentifier: S3_BUCKET_PUBLIC_READ_PROHIBITED
    Scope:
      ComplianceResourceTypes:
        - AWS::S3::Bucket
```

**Detect Public Write Access Setting for Log Archive**

This control detects whether public write access is enabled to the Amazon S3 buckets in the log archive shared account. This control does not change the status of the account. This is a detective control with mandatory guidance. By default, this control is enabled on the **Security OU**.

The artifact for this control is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 2010-09-09
```
Description: Configure AWS Config rules to check that your S3 buckets do not allow public access
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForS3PublicWrite:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks that your S3 buckets do not allow public write access. If an S3 bucket policy or bucket ACL allows public write access, the bucket is noncompliant.
      Source:
        Owner: AWS
        SourceIdentifier: S3_BUCKET_PUBLIC_WRITE_PROHIBITED
      Scope:
        ComplianceResourceTypes:
          - AWS::S3::Bucket

Disallow Configuration Changes to CloudTrail

This control prevents configuration changes to CloudTrail in your landing zone. This is a preventive control with mandatory guidance. By default, this control is enabled on all OUs.

The artifact for this control is the following SCP.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRCLOUDTRAILENABLED",
      "Effect": "Deny",
      "Action": [
        "cloudtrail:DeleteTrail",
        "cloudtrail:PutEventSelectors",
        "cloudtrail:StopLogging",
        "cloudtrail:UpdateTrail"
      ],
      "Resource": ["arn:aws:cloudtrail:*::*:trail/aws-controltower-*"],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
  ]
}
```

Integrate CloudTrail Events with Amazon CloudWatch Logs

This control performs real-time analysis of activity data by sending CloudTrail events to CloudWatch Logs log files. This is a preventive control with mandatory guidance. By default, this control is enabled on all OUs.

The artifact for this control is the following SCP.

```json
[
  "Version": "2012-10-17",
  "Statement": [
    {
      
```
Enable CloudTrail in All Available Regions

This control enables CloudTrail in all available AWS Regions. This is a preventive control with mandatory guidance. By default, this control is enabled in all OUs.

The artifact for this control is the following SCP.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCLOUDTRAILENABLED",
            "Effect": "Deny",
            "Action": [
                "cloudtrail:DeleteTrail",
                "cloudtrail:PutEventSelectors",
                "cloudtrail:StopLogging",
                "cloudtrail:UpdateTrail"
            ],
            "Resource": ["arn:aws:cloudtrail:*:*:trail/aws-controltower-*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Enable Integrity Validation for CloudTrail Log File

This control enables integrity validation for the CloudTrail log file in all accounts and OUs. It protects the integrity of account activity logs using CloudTrail log file validation, which creates a digitally signed digest file that contains a hash of each log that CloudTrail writes to Amazon S3. This is a preventive control with mandatory guidance. By default, this control is enabled in all OUs.

The artifact for this control is the following SCP.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCLOUDTRAILENABLED",
            "Effect": "Deny",
            "Action": [
                "cloudtrail:DeleteTrail",
                "cloudtrail:PutEventSelectors",
                "cloudtrail:StopLogging",
                "cloudtrail:UpdateTrail"
            ],
            "Resource": ["arn:aws:cloudtrail:*:*:trail/aws-controltower-*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```
Disallow Changes to Amazon CloudWatch Set Up by AWS Control Tower

This control disallows changes to Amazon CloudWatch; as it was configured by AWS Control Tower when you set up your landing zone. This is a preventive control with mandatory guidance. By default, this control is enabled in all OUs.

The artifact for this control is the following SCP.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCLOUDWATCHEVENTPOLICY",
            "Effect": "Deny",
            "Action": [
                "events:PutRule",
                "events:PutTargets",
                "events:RemoveTargets",
                "events:DisableRule",
                "events:DeleteRule"
            ],
            "Resource": [
                "arn:aws:events:*:*:rule/aws-controltower-*"
            ],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Disallow Changes to Tags Created by AWS Control Tower for AWS Config Resources

This control prevents changes to the tags that AWS Control Tower created when you set up your landing zone, for AWS Config resources that collect configuration and compliance data. It denies any TagResource and UntagResource operation for aggregation authorizations tagged by AWS Control Tower. This is a preventive control with mandatory guidance. By default, this control is enabled in all OUs.
The artifact for this control is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCONFIGRULETAGSPOLICY",
            "Effect": "Deny",
            "Action": [
                "config:TagResource",
                "config:UntagResource"
            ],
            "Resource": ["*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "aws-control-tower"
                }
            }
        }
    ]
}
```

**Disallow Configuration Changes to AWS Config**

This control prevents configuration changes to AWS Config. It ensures that AWS Config records resource configurations in a consistent manner by disallowing AWS Config settings changes. This is a preventive control with mandatory guidance. By default, this control is enabled in all OUs.

The artifact for this control is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCONFIGENABLED",
            "Effect": "Deny",
            "Action": [
                "config:DeleteConfigurationRecorder",
                "config:DeleteDeliveryChannel",
                "config:DeleteRetentionConfiguration",
                "config:PutConfigurationRecorder",
                "config:PutDeliveryChannel",
                "config:PutRetentionConfiguration",
                "config:StopConfigurationRecorder"
            ],
            "Resource": ["*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

**Enable AWS Config in All Available Regions**

This control enables AWS Config in all available AWS Regions. This is a preventive control with mandatory guidance. By default, this control is enabled in all OUs.
The artifact for this control is the following SCP.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRCONFIGENABLED",
         "Effect": "Deny",
         "Action": [
            "config:DeleteConfigurationRecorder",
            "config:DeleteDeliveryChannel",
            "config:DeleteRetentionConfiguration",
            "config:PutConfigurationRecorder",
            "config:PutDeliveryChannel",
            "config:PutRetentionConfiguration",
            "config:StopConfigurationRecorder"
         ],
         "Resource": ["*"]
      },
      "Condition": {
         "ArnNotLike": {
            "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
         }
      }
   }
}
```

Disallow Changes to AWS Config Rules Set Up by AWS Control Tower

This control disallows changes to AWS Config Rules that were implemented by AWS Control Tower when the landing zone was set up. This is a preventive control with mandatory guidance. By default, this control is enabled in all OUs.

The artifact for this control is the following SCP.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRCONFIGRULEPOLICY",
         "Effect": "Deny",
         "Action": [
            "config:PutConfigRule",
            "config:DeleteConfigRule",
            "config:DeleteEvaluationResults",
            "config:DeleteConfigurationAggregator",
            "config:PutConfigurationAggregator"
         ],
         "Resource": ["*"]
      },
      "Condition": {
         "ArnNotLike": {
            "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
         },
         "StringEquals": {
            "aws:ResourceTag/aws-control-tower": "managed-by-control-tower"
         }
      }
   }
}
```
Disallow Changes to AWS IAM Roles Set Up by AWS Control Tower and AWS CloudFormation

This control disallows changes to the AWS IAM roles that AWS Control Tower created when the landing zone was set up. This is a preventive control with mandatory guidance. By default, this control is enabled in all OUs.

Control update

An updated version has been released for the mandatory control AWS-GR_IAM_ROLE_CHANGE_PROHIBITED.

This change to the control is required because accounts in OUs that are being enrolled into AWS Control Tower must have the AWSControlTowerExecution role enabled. The previous version of the control prevents this role from being created.

AWS Control Tower updated the existing control to add an exception so that AWS CloudFormation StackSets can create the AWSControlTowerExecution role. As a second measure, this new control protects the StackSets role to prevent principals in the child account from gaining access.

The new control version performs the following actions, in addition to all actions provided in the previous version:

- Allows the stacksets-exec-* role (owned by AWS CloudFormation) to perform actions on IAM roles that were created by AWS Control Tower.
- Prevents changes to any IAM role in child accounts, where the IAM role name matches the pattern stacksets-exec-*.

The update to the control version affects your OUs and accounts as follows:

- If you extend governance to an OU, that incoming OU receives the updated version of the control as part of the registration process. You do not need to update your landing zone to get the latest version for this OU. AWS Control Tower applies the latest version automatically to OUs that register.
- If you update or repair your landing zone at any time after this release, your control will be updated to this version for future provisioning.
- OUs created in or registered with AWS Control Tower before this release date, and which are part of a landing zone that has not been repaired or updated after the release date, will continue to operate with the old version of the control, which blocks the creation of the AWSControlTowerExecution role.
- One consequence of this control update is that your OUs can be functioning with different versions of the control. Update your landing zone to apply the updated version of the control to your OUs uniformly.

The artifact of the updated control is the following SCP.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRIAMROLEPOLICY",
            "Effect": "Deny",
            "Action": [
                "iam:AttachRolePolicy"
            ]
        }
    ]
}
```
The former artifact for this control is the following SCP.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRIAMROLEPOLICY",
      "Effect": "Deny",
      "Action": [
        "iam:AttachRolePolicy",
        "iam:CreateRole",
        "iam:DeleteRole",
        "iam:DeleteRolePermissionsBoundary",
        "iam:DeleteRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePermissionsBoundary",
        "iam:PutRolePolicy",
        "iam:UpdateAssumeRolePolicy",
        "iam:UpdateRole",
        "iam:UpdateRoleDescription"
      ],
      "Resource": [
        "arn:aws:iam::*:role/aws-controltower-*",
        "arn:aws:iam::*:role/*AWSControlTower*",
        "arn:aws:iam::*:role/stacksets-exec-*"  #this line is new
      ],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN": ["arn:aws:iam::*:role/AWSControlTowerExecution",
                                 "arn:aws:iam::*:role/stacksets-exec-*"  #this line is new
                      ]
        }
      }
    }
  ]
}
```
Disallow Changes to AWS Lambda Functions Set Up by AWS Control Tower

This control disallows changes to AWS Lambda functions set up by AWS Control Tower. This is a preventive control with mandatory guidance. By default, this control is enabled in all OUs.

The artifact for this control is the following SCP.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRLAMBDAFUNCTIONPOLICY",
         "Effect": "Deny",
         "Action": [
            "lambda:AddPermission",
            "lambda:CreateEventSourceMapping",
            "lambda:CreateFunction",
            "lambda:DeleteEventSourceMapping",
            "lambda:DeleteFunction",
            "lambda:DeleteFunctionConcurrency",
            "lambda:PutFunctionConcurrency",
            "lambda:RemovePermission",
            "lambda:UpdateEventSourceMapping",
            "lambda:UpdateFunctionConfiguration"
         ],
         "Resource": [
            "arn:aws:lambda:*:*:function:aws-controltower-*"
         ],
         "Condition": {
            "ArnNotLike": {
               "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
            }
         }
      }
   ]
}
```

Disallow Changes to Amazon SNS Set Up by AWS Control Tower

This control disallows changes to Amazon SNS set up by AWS Control Tower. It protects the integrity of Amazon SNS notification settings for your landing zone. This is a preventive control with mandatory guidance. By default, this control is enabled in all OUs.

The artifact for this control is the following SCP.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRSNSTOPICPOLICY",
         "Effect": "Deny",
         "Action": [
            "sns:AddPermission",
            "sns:CreateTopic",
            "sns:DeleteTopic",
            "sns:RemovePermission",
            "sns:SetTopicAttributes"
         ],
         "Resource": [
            "arn:aws:sns:*:*:topic/aws-controltower-*"
         ],
         "Condition": {
            "ArnNotLike": {
               "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
            }
         }
      }
   ]
}
```
Disallow Changes to Amazon SNS Subscriptions Set Up by AWS Control Tower

This control disallows changes to Amazon SNS subscriptions set up by AWS Control Tower. It protects the integrity of Amazon SNS subscriptions settings for your landing zone, to trigger notifications for AWS Config Rules compliance changes. This is a preventive control with mandatory guidance. By default, this control is enabled in all OUs.

The artifact for this control is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRSNSSUBSCRIPTIONPOLICY",
      "Effect": "Deny",
      "Action": [
        "sns:Subscribe",
        "sns:Unsubscribe"
      ],
      "Resource": [
        "arn:aws:sns:*:*:aws-controltower-SecurityNotifications"
      ],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
  ]
}
```

Detect whether shared accounts under the Security organizational unit have AWS CloudTrail or CloudTrail Lake enabled

This control detects whether shared accounts under the Security organizational unit have AWS CloudTrail or CloudTrail Lake enabled. The rule is NON_COMPLIANT if either CloudTrail or CloudTrail Lake is not enabled in a shared account. This is a detective control with mandatory guidance. By default, this control is enabled on the Security OU.

The artifact for this control is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to detect whether an account has AWS CloudTrail or CloudTrail Lake enabled.
```
Proactive controls

Proactive controls are optional controls implemented with AWS CloudFormation hooks. Proactive controls fall into three main Categories.

These controls are referred to as proactive because they check your resources – before the resources are deployed – to determine whether the new resources will comply with the controls that are activated in your environment.

In the AWS Control Tower console, you can view the controls in groups according to their assigned categories, which are:

- **Control objectives**: Specific purposes for implementing controls in your environment.
- **Frameworks**: Industry-standard compliance frameworks.
- **Services**: The AWS services that the control may govern.

In this reference guide, the proactive controls are categorized according to their associated AWS services.

**Note**
You must apply an elective, SCP-based control with the identifier CT.CLOUDFORMATION.PR.1 before you can activate proactive controls on an OU. See Disallow management of resource types, modules, and hooks within the AWS CloudFormation registry (p. 1375). If this SCP is not activated, you’ll see an error message directing you to enable this control as a prerequisite, or showing it as a dependency for other proactive controls.

**Behavior of proactive controls**

Proactive controls check resources whenever those resources are created or updated by means of AWS CloudFormation stack operations. Specifically, these proactive controls are implemented as preCreate and preUpdate AWS CloudFormation hook handlers. As a consequence, these controls may not affect requests that are made directly to services through the AWS console, through AWS APIs, or through other means such as AWS SDKs, or other Infrastructure-as-Code (IaC) tools. For more information about when preCreate and preUpdate hooks operate, see AWS CloudFormation hooks.

When you follow an example template to set up a test for a proactive control in your environment, be aware that the template is created to test one specific control only. Other controls may not receive a PASS rating for that template. This behavior is expected. We recommend that you test proactive controls individually before you enable them in your environment.

**Note**
It is important to know that some proactive controls in AWS Control Tower do not operate in certain AWS Regions where AWS Control Tower is available, because those Regions do not
support the required underlying functionality for AWS CloudFormation hooks. As a result, when you deploy a proactive control through AWS Control Tower, the control may not be operating in all Regions that you govern with AWS Control Tower.
You can view the Regions for each proactive control in the AWS Control Tower console.

Topics

- Amazon API Gateway controls (p. 232)
- AWS Certificate Manager controls (p. 259)
- AWS AppSync controls (p. 264)
- Amazon Athena controls (p. 269)
- Amazon CloudFront controls (p. 280)
- AWS CloudTrail controls (p. 352)
- Amazon CloudWatch controls (p. 374)
- AWS CodeBuild controls (p. 389)
- AWS Database Migration Service (AWS DMS) controls (p. 435)
- Amazon DocumentDB controls (p. 438)
- Amazon DynamoDB controls (p. 446)
- DynamoDB Accelerator controls (p. 454)
- AWS Elastic Beanstalk controls (p. 464)
- Amazon Elastic Compute Cloud (Amazon EC2) controls (p. 494)
- Amazon Elastic Compute Cloud (Amazon EC2) Auto Scaling controls (p. 568)
- Amazon ElastiCache controls (p. 604)
- Amazon Elastic Container Registry controls (p. 636)
- Amazon Elastic Container Service controls (p. 646)
- Amazon Elastic File System controls (p. 706)
- Amazon Elastic Kubernetes Service (EKS) controls (p. 721)
- Elastic Load Balancing controls (p. 734)
- Amazon Elastic Map Reduce (Amazon EMR) controls (p. 827)
- AWS Glue controls (p. 847)
- Amazon GuardDuty controls (p. 852)
- AWS Identity and Access Management (IAM) controls (p. 856)
- AWS Key Management Service (AWS KMS) controls (p. 892)
- Amazon Kinesis controls (p. 907)
- AWS Lambda controls (p. 912)
- Amazon Neptune controls (p. 941)
- AWS Network Firewall controls (p. 957)
- Amazon OpenSearch controls (p. 977)
- Amazon Relational Database Service (Amazon RDS) controls (p. 1069)
- Amazon Redshift controls (p. 1201)
- Amazon Simple Storage Service (Amazon S3) controls (p. 1239)
- Amazon SageMaker controls (p. 1280)
- Amazon Simple Queue Service (Amazon SQS) controls (p. 1294)
- AWS Step Functions controls (p. 1303)
- AWS WAF regional controls (p. 1317)
- AWS WAF controls (p. 1324)
AWS Control Tower User Guide
Proactive controls

- AWS WAFV2 controls (p. 1332)

Amazon API Gateway controls

Topics
- [CT.APIGATEWAY.PR.1] Require an Amazon API Gateway REST and WebSocket API to have logging activated (p. 232)
- [CT.APIGATEWAY.PR.2] Require an Amazon API Gateway REST API stage to have AWS X-Ray tracing activated (p. 238)
- [CT.APIGATEWAY.PR.3] Require that an Amazon API Gateway REST API stage has encryption at rest configured for cache data (p. 242)
- [CT.APIGATEWAY.PR.4] Require an Amazon API Gateway V2 stage to have access logging activated (p. 249)
- [CT.APIGATEWAY.PR.5] Require Amazon API Gateway V2 Websocket and HTTP routes to specify an authorization type (p. 253)

[CT.APIGATEWAY.PR.1] Require an Amazon API Gateway REST and WebSocket API to have logging activated

This control checks whether all methods in Amazon API Gateway stage have execution logging configured.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ApiGateway::Stage
- **AWS CloudFormation guard rule:** CT.APIGATEWAY.PR.1 rule specification (p. 233)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.APIGATEWAY.PR.1 rule specification (p. 233)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.APIGATEWAY.PR.1 example templates (p. 236)

Explanation

Amazon API Gateway REST or WebSocket API stages should have relevant logs enabled. API Gateway REST and WebSocket API execution logging provides detailed records of requests made to API Gateway REST and WebSocket API stages. The stages include API integration backend responses, Lambda authorizer responses, and the requestId for AWS integration endpoints.

Usage considerations

- This control requires Amazon API Gateway stages to configure execution logging for all methods and resources (HttpMethod of * and ResourcePath of /*).

Remediation for rule failure

Configure execution logging on Amazon API Gateway stages with a MethodSetting that sets LoggingLevel to ERROR or INFO for all methods (HttpMethod of * and ResourcePath of /*). Ensure that you do not set LoggingLevel to OFF for any method setting.
The examples that follow show how to implement this remediation.

**Amazon API Gateway Stage - Example**

Amazon API Gateway stage configured with **error** level execution logging for all methods and resources. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ApiGatewayStage": {
        "Type": "AWS::ApiGateway::Stage",
        "Properties": {
            "StageName": "Sample",
            "Description": "Sample Stage",
            "RestApiId": {
                "Ref": "RestApi"
            },
            "DeploymentId": {
                "Ref": "Deployment"
            },
            "MethodSettings": [
                {
                    "ResourcePath": "/*",
                    "HttpMethod": "*",
                    "LoggingLevel": "ERROR"
                }
            ]
        }
    }
}
```

**YAML example**

```yaml
ApiGatewayStage:
  Type: AWS::ApiGateway::Stage
  Properties:
    StageName: Sample
    Description: Sample Stage
    RestApiId: !Ref 'RestApi'
    DeploymentId: !Ref 'Deployment'
    MethodSettings:
      - ResourcePath: /*
        HttpMethod: '*'
        LoggingLevel: ERROR
```

**CT.APIGATEWAY.PR.1 rule specification**

```
# ######################################################################
## Rule Specification ##
# ######################################################################

# Rule Identifier:
# api_gw_v1_execution_logging_enabled_check
#
# Description:
```
This control checks whether all methods in Amazon API Gateway stage have execution logging configured.

Reports on:
- AWS::ApiGateway::Stage

Evaluates:
- AWS CloudFormation, AWS CloudFormation hook

Rule Parameters:
- None

Scenarios:

Scenario: 1
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document does not contain any API Gateway stage resources
- Then: SKIP

Scenario: 2
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document contains an API Gateway stage resource
- And: In the stage resource, 'MethodSettings' is not present or is provided and is an empty list.
- Then: FAIL

Scenario: 3
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document contains an API Gateway stage resource
- And: In the stage resource, Execution Logging is not configured for all HTTP Methods and API resources (In 'MethodSettings', 'LoggingLevel' is omitted, or not set to 'ERROR' or 'INFO', for 'HttpMethod' of '*' and 'ResourcePath' of '/*')
- Then: FAIL

Scenario: 4
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document contains an API Gateway stage resource
- And: In the stage resource, Execution Logging is configured for all HTTP Methods and API resources (In 'MethodSettings', 'LoggingLevel' is set to 'ERROR' or 'INFO', for 'HttpMethod' of '*' and 'ResourcePath' of '/*')
- And: 'LoggingLevel' has been set to 'OFF' for any other Method Setting
- Then: FAIL

Scenario: 5
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document contains an API Gateway stage resource
- And: In the stage resource, Execution Logging is configured for all HTTP Methods and API resources (In 'MethodSettings', 'LoggingLevel' is set to 'ERROR' or 'INFO', for 'HttpMethod' of '*' and 'ResourcePath' of '/*')
- And: 'LoggingLevel' has not been provided or set to 'ERROR' or 'INFO' for all other Method Settings
- Then: PASS

Constants

let API_GW_STAGE_TYPE = "AWS::ApiGateway::Stage"
let INPUT_DOCUMENT = this
let VALID_LOG_LEVELS = [ "ERROR", "INFO" ]
# Assignments

let api_gateway_stages = Resources.*[ Type == %API_GW_STAGE_TYPE ]

# Primary Rules

# rule api_gw_v1_execution_logging_enabled_check when is_cfn_template(%INPUT_DOCUMENT) { %api_gateway_stages not empty { check(%api_gateway_stages.Properties) << [CT.APIGATEWAY.PR.1]: Require an Amazon API Gateway REST and WebSocket API to have logging activated

[Fix]: Configure execution logging on Amazon API Gateway stages with a 'MethodSetting' that sets 'LoggingLevel' to 'ERROR' or 'INFO' for all methods ('HttpMethod' of '*' and 'ResourcePath' of '/*'). Ensure that you do not set 'LoggingLevel' to 'OFF' for any method setting.

} }

rule api_gw_v1_execution_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %API_GW_STAGE_TYPE) { check(%INPUT_DOCUMENT.%API_GW_STAGE_TYPE.resourceProperties) << [CT.APIGATEWAY.PR.1]: Require an Amazon API Gateway REST and WebSocket API to have logging activated

[Fix]: Configure execution logging on Amazon API Gateway stages with a 'MethodSetting' that sets 'LoggingLevel' to 'ERROR' or 'INFO' for all methods ('HttpMethod' of '*' and 'ResourcePath' of '/*'). Ensure that you do not set 'LoggingLevel' to 'OFF' for any method setting.

} }

# Parameterized Rules

# rule check(api_gateway_stage) { %api_gateway_stage { # Scenario 2

MethodSettings exists
MethodSettings is_list
MethodSettings not empty

# Scenario 3
# At least one wildcard entry exists with valid logging enabled
some MethodSettings[*] { HttpMethod exists
ResourcePath exists
LoggingLevel exists

HttpMethod == "*"
ResourcePath == "/*"
LoggingLevel in %VALID_LOG_LEVELS
}

# Scenario 4, 5
# When other methods explicitly set/override logging settings, ensure that logging is not disabled
MethodSettings[*] { when LoggingLevel exists {
LoggingLevel in %VALID_LOG_LEVELS
}
}
} }
# Utility Rules

## rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

## rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

### CT.APIGATEWAY.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```json
Resources:
  RestApi:
    Type: AWS::ApiGateway::RestApi
    Properties:
      Name: ExampleRestApi
  GetMethod:
    DependsOn: PutMethod
    Type: AWS::ApiGateway::Method
    Properties:
      HttpMethod: GET
      RestApiId:
        Ref: RestApi
      ResourceId:
        Fn::GetAtt:
          - "RestApi"
          - "RootResourceId"
      AuthorizationType: NONE
      MethodResponses:
        - StatusCode: "200"
      Integration:
        Type: MOCK
  PutMethod:
    Type: AWS::ApiGateway::Method
    Properties:
      HttpMethod: PUT
      RestApiId:
        Ref: RestApi
      ResourceId:
        Fn::GetAtt:
          - "RestApi"
          - "RootResourceId"
      AuthorizationType: NONE
      MethodResponses:
        - StatusCode: "200"
      Integration:
        Type: MOCK
  Deployment:
    DependsOn: GetMethod
    Type: 'AWS::ApiGateway::Deployment'
    Properties:
```

236
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  RestApi:
    Type: AWS::ApiGateway::RestApi
    Properties:
      Name: ExampleRestApi
    GetMethod:
      DependsOn: PutMethod
      Type: AWS::ApiGateway::Method
      Properties:
        HttpMethod: GET
        RestApiId:
          Ref: RestApi
        ResourceId:
          Fn::GetAtt:
            - "RestApi"
            - "RootResourceId"
        AuthorizationType: NONE
        MethodResponses:
          - StatusCode: "200"
        Integration:
          Type: MOCK
    PutMethod:
      Type: AWS::ApiGateway::Method
      Properties:
        HttpMethod: PUT
        RestApiId:
          Ref: RestApi
        ResourceId:
          Fn::GetAtt:
            - "RestApi"
            - "RootResourceId"
        AuthorizationType: NONE
        MethodResponses:
          - StatusCode: "200"
        Integration:
          Type: MOCK
  Deployment:
    DependsOn: GetMethod
    Type: 'AWS::ApiGateway::Deployment'
    Properties:
      RestApiId:
        Ref: RestApi
[CT.APIGATEWAY.PR.2] Require an Amazon API Gateway REST API stage to have AWS X-Ray tracing activated

This control ensures that AWS X-Ray tracing is enabled on Amazon API Gateway REST APIs.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ApiGateway::Stage
- **AWS CloudFormation guard rule:** [CT.APIGATEWAY.PR.2 rule specification (p. 239)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.APIGATEWAY.PR.2 rule specification (p. 239)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.APIGATEWAY.PR.2 example templates (p. 241)]

**Explanation**

AWS X-Ray active tracing enables a more rapid response to performance changes in the underlying infrastructure. Changes in performance could result in a lack of availability of the API. X-Ray active tracing provides real-time metrics of user requests that flow through your API Gateway REST API operations and connected services.

**Remediation for rule failure**

Set `TracingEnabled` to `true`.

The examples that follow show how to implement this remediation.

**Amazon API Gateway Stage - Example**

Amazon API Gateway stage configured with AWS X-Ray tracing enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ApiGatewayStage": {
        "Type": "AWS::ApiGateway::Stage",
        "Properties": {
            "StageName": "Sample",
            "Description": "Sample Stage",
            "TracingEnabled": true,
```
"RestApiId": {
    "Ref": "RestApi"
},
"DeploymentId": {
    "Ref": "Deployment"
}
}
}

YAML example

ApiGatewayStage:
  Type: AWS::ApiGateway::Stage
  Properties:
    StageName: Sample
    Description: Sample Stage
    TracingEnabled: true
    RestApiId: !Ref 'RestApi'
    DeploymentId: !Ref 'Deployment'

CT.APIGATEWAY.PR.2 rule specification

# ######################################################################
# Rule Specification  ##
# ######################################################################
#
# Rule Identifier:
# api_gw_xray_enabled_check
#
# Description:
# This control ensures that AWS X-Ray tracing is enabled on Amazon API Gateway REST APIs.
#
# Reports on:
# AWS::ApiGateway::Stage
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document does not contain any API Gateway stage resources
#   Then: SKIP
# Scenario: 2
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains an API Gateway stage resource
#   And: 'TracingEnabled' is not present on the API Gateway stage
#   Then: FAIL
# Scenario: 3
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains an API Gateway stage resource

239
And: 'TracingEnabled' is present on the API Gateway stage and is set to bool(false)
Then: FAIL
Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document contains an API Gateway stage resource
And: 'TracingEnabled' is present on the API Gateway stage and is set to bool(true)
Then: PASS

Constants
let API_GW_STAGE_TYPE = "AWS::ApiGateway::Stage"
let INPUT_DOCUMENT = this

Assignments
let api_gateway_stages = Resources.[ Type == %API_GW_STAGE_TYPE ]

Primary Rules
rule api_gw_xray_enabled_check when is_cfn_template(%INPUT_DOCUMENT) {
%api_gateway_stages not empty {
check(%api_gateway_stages.Properties)
<<
[CT.APIGATEWAY.PR.2]: Require an Amazon API Gateway REST API stage to have AWS X-Ray tracing activated
[FIX]: Set 'TracingEnabled' to 'true'.
>>
}
}
rule api_gw_xray_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %API_GW_STAGE_TYPE) {
check(%INPUT_DOCUMENT.%API_GW_STAGE_TYPE.resourceProperties)
<<
[CT.APIGATEWAY.PR.2]: Require an Amazon API Gateway REST API stage to have AWS X-Ray tracing activated
[FIX]: Set 'TracingEnabled' to 'true'.
>>
}

Parameterized Rules
rule check(api_gateway_stage) {
%api_gateway_stage {
# Scenario 2, 3, 4
TracingEnabled exists
TracingEnabled == true
}
}

Utility Rules
rule is_cfn_template(doc) {
%doc {
AWSTemplateFormatVersion exists or Resources exists
}
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
%doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.APIGATEWAY.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  RestApi:
    Type: AWS::ApiGateway::RestApi
    Properties:
      Name: Testing
  GetMethod:
    DependsOn: PutMethod
    Type: AWS::ApiGateway::Method
    Properties:
      HttpMethod: GET
      RestApiId:
        Ref: RestApi
      ResourceId:
        Fn::GetAtt:
          - "RestApi"
          - "RootResourceId"
      AuthorizationType: NONE
      MethodResponses:
        - StatusCode: "200"
      Integration:
        Type: MOCK
  PutMethod:
    Type: AWS::ApiGateway::Method
    Properties:
      HttpMethod: PUT
      RestApiId:
        Ref: RestApi
      ResourceId:
        Fn::GetAtt:
          - "RestApi"
          - "RootResourceId"
      AuthorizationType: NONE
      MethodResponses:
        - StatusCode: "200"
      Integration:
        Type: MOCK
  Deployment:
    DependsOn: GetMethod
    Type: 'AWS::ApiGateway::Deployment'
    Properties:
      RestApiId:
        Ref: RestApi
      ApiGatewayStage:
        Type: AWS::ApiGateway::Stage
        Properties:
          StageName: Dev
          Description: Dev Stage
          TracingEnabled: true
          RestApiId:
            Ref: RestApi
          DeploymentId:
            Ref: Deployment
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
RestApi:
  Type: AWS::ApiGateway::RestApi
  Properties:
    Name: Testing
GetMethod:
  DependsOn: PutMethod
  Type: AWS::ApiGateway::Method
  Properties:
    HttpMethod: GET
    RestApiId:
      Ref: RestApi
    ResourceId:
      Fn::GetAtt:
        - "RestApi"
        - "RootResourceId"
    AuthorizationType: NONE
    MethodResponses:
      - StatusCode: "200"
    Integration:
      Type: MOCK
PutMethod:
  Type: AWS::ApiGateway::Method
  Properties:
    HttpMethod: PUT
    RestApiId:
      Ref: RestApi
    ResourceId:
      Fn::GetAtt:
        - "RestApi"
        - "RootResourceId"
    AuthorizationType: NONE
    MethodResponses:
      - StatusCode: "200"
    Integration:
      Type: MOCK
Deployment:
  DependsOn: GetMethod
  Type: 'AWS::ApiGateway::Deployment'
  Properties:
    RestApiId:
      Ref: RestApi

[CT.APIGATEWAY.PR.3] Require that an Amazon API Gateway REST API stage has encryption at rest configured for cache data

This control checks whether an Amazon API Gateway REST API stage that has caching enabled also encrypts the caches.
• **Control objective:** Encrypt data at rest
• **Implementation:** AWS CloudFormation Guard Rule
• **Control behavior:** Proactive
• **Resource types:** `AWS::ApiGateway::Stage`
• **AWS CloudFormation guard rule:** [CT.APIGATEWAY.PR.3 rule specification (p. 244)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.APIGATEWAY.PR.3 rule specification (p. 244)]
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.APIGATEWAY.PR.3 example templates (p. 247)]

**Explanation**

Encrypting data at rest reduces the risk that data stored on disk may be accessible by a user not authenticated to AWS. It adds another set of access controls to limit unauthorized users' ability to obtain the data. For example, API permissions are required to decrypt the data before it can be read.

For an added layer of security, API Gateway REST API caches should be encrypted at rest.

**Usage considerations**

- This control applies only to API Gateway stage resources with cache clustering enabled.
- Where cache clustering is enabled, this control requires cache encryption to be enabled for all resources and methods by specifying a `MethodSetting` entry with an `HttpMethod` of `*` and `ResourcePath` of `/*`.

**Remediation for rule failure**

Configure encryption on API Gateway caches with a `MethodSetting` that sets `CacheDataEncrypted` to true for all methods (`HttpMethod` of `*` and `ResourcePath` of `/*`). Ensure that you do not set `CacheDataEncrypted` to false for any method setting.

The examples that follow show how to implement this remediation.

**API Gateway stage examples**

This example shows the API Gateway stage configured to encrypt cache data for all methods (`HttpMethod` of `*` and `ResourcePath` of `/*`). The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ApiGatewayStage": {
        "Type": "AWS::ApiGateway::Stage",
        "Properties": {
            "StageName": "Dev",
            "Description": "Development Stage",
            "CacheClusterEnabled": true,
            "CacheClusterSize": 0.5,
            "RestApiId": {
                "Ref": "RestApi"
            },
            "DeploymentId": {
                "Ref": "Deployment"  
            }  
    }
}
```
YAML example

```yaml
ApiGatewayStage:
  Type: AWS::ApiGateway::Stage
  Properties:
    StageName: Dev
    Description: Development Stage
    CacheClusterEnabled: true
    CacheClusterSize: 0.5
    RestApiId: !Ref 'RestApi'
    DeploymentId: !Ref 'Deployment'
    MethodSettings:
      - ResourcePath: /*
        HttpMethod: '*'
        CacheDataEncrypted: true
      - ResourcePath: /
        HttpMethod: POST
```

CT.APIGATEWAY.PR.3 rule specification

```bash
# ###################################################################
##       Rule Specification        ##
###################################################################
# # Rule Identifier:
#   api_gw_cache_encrypted_check
# # Description:
#   This rule checks whether Amazon API Gateway REST API stages that have caching enabled also encrypt the caches.
# # Reports on:
#   AWS::ApiGateway::Stage
# # Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# # Rule Parameters:
#   None
# # Scenarios:
#   Scenario: 1
```
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document does not contain any Amazon API Gateway stage resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon API Gateway stage resource
# And: 'CacheClusterEnabled' is not set, or is set to bool(false) on the API Gateway stage resource
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon API Gateway stage resource
# And: 'CacheClusterEnabled' is set to bool(true) on the API Gateway stage resource
# And: In the stage resource, 'MethodSettings' is not present or is provided and is an empty list.
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon API Gateway stage resource
# And: 'CacheClusterEnabled' is set to bool(true) on the API Gateway stage resource
# And: In the stage resource, cache data encryption is not enabled for all HTTP methods and API resources (In
# 'MethodSettings', 'CacheDataEncrypted' is omitted or set to bool(false) for 'HttpMethod' of '*' and 'ResourcePath' of '/*')
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon API Gateway stage resource
# And: 'CacheClusterEnabled' is set to bool(true) on the API Gateway stage resource
# And: In the stage resource, cache data encryption is configured for all 'MethodSettings' (CacheDataEncrypted is
# bool(true) for 'HttpMethod' of '*' and 'ResourcePath' of '/*')
# And: 'CacheDataEncrypted' has been set to bool(false) for any other method settings
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon API Gateway stage resource
# And: 'CacheClusterEnabled' is set to bool(true) on the API Gateway stage resource
# And: In the stage resource cache data encryption is configured for all 'MethodSettings' (CacheDataEncrypted is
# bool(true) for 'HttpMethod' of '*' and 'ResourcePath' of '/*')
# And: 'CacheDataEncrypted' has not been provided or set to bool(true) for all other method settings
# Then: PASS

# Constants
# let API_GW_STAGE_TYPE = "AWS::ApiGateway::Stage"
let INPUT_DOCUMENT = this

# Assignments
# let api_gateway_stages = Resources.*[ Type == %API_GW_STAGE_TYPE ]

# Primary Rules
# rule api_gw_cache_encrypted_check when is_cfn_template(%INPUT_DOCUMENT)
#  %api_gateway_stages not empty {
#   check(%api_gateway_stages.Properties)
#   <<
#   [CT.APIGATEWAY.PR.3]: Require that an Amazon API Gateway REST API stage has encryption at rest configured for cache data
[FIX]: Configure encryption on API Gateway caches with a 'MethodSetting' that sets 'CacheDataEncrypted' to true for all methods ('HttpMethod' of '*' and 'ResourcePath' of '/ *'). Ensure that you do not set 'CacheDataEncrypted' to false for any method setting.

```python
rule api_gw_cache_encrypted_check when is_cfn_hook(%INPUT_DOCUMENT, %API_GW_STAGE_TYPE) {
    check(%INPUT_DOCUMENT.%API_GW_STAGE_TYPE.resourceProperties)
}
```

[CT.APIGATEWAY.PR.3]: Require that an Amazon API Gateway REST API stage has encryption at rest configured for cache data

[FIX]: Configure encryption on API Gateway caches with a 'MethodSetting' that sets 'CacheDataEncrypted' to true for all methods ('HttpMethod' of '*' and 'ResourcePath' of '/ *'). Ensure that you do not set 'CacheDataEncrypted' to false for any method setting.

```python
rule check(api_gateway_stage) {
    %api_gateway_stage [
        CacheClusterEnabled exists
        CacheClusterEnabled == true
    ] {
        # Scenario 2, 3, 4, 6
        cache_encrypted(this)
    }
}
```

```python
rule cache_encrypted(api_gateway_stage) {
    %api_gateway_stage {
        MethodSettings exists
        MethodSettings is_list
        MethodSettings not empty
        some MethodSettings[*] {
            HttpMethod exists
            ResourcePath exists
            CacheDataEncrypted exists
            HttpMethod == "*"
            ResourcePath == "/*"
            CacheDataEncrypted == true
        }
        MethodSettings[*] {
            when CacheDataEncrypted exists {
                CacheDataEncrypted == true
            }
        }
    }
}
```

# Utility Rules

```python
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
```

```python
rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```
CT.APIGATEWAY.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
RestApi:
  Type: AWS::ApiGateway::RestApi
  Properties:
    Name: ExampleRestApi
GetMethod:
  DependsOn: PutMethod
  Type: AWS::ApiGateway::Method
  Properties:
    HttpMethod: GET
    RestApiId:
      Ref: RestApi
    ResourceId:
      Fn::GetAtt:
        - RestApi
        - "RootResourceId"
    AuthorizationType: NONE
    MethodResponses:
      - StatusCode: "200"
    Integration:
      Type: MOCK
PutMethod:
  Type: AWS::ApiGateway::Method
  Properties:
    HttpMethod: PUT
    RestApiId:
      Ref: RestApi
    ResourceId:
      Fn::GetAtt:
        - RestApi
        - "RootResourceId"
    AuthorizationType: NONE
    MethodResponses:
      - StatusCode: "200"
    Integration:
      Type: MOCK
Deployment:
  DependsOn: GetMethod
  Type: 'AWS::ApiGateway::Deployment'
  Properties:
    RestApiId:
      Ref: RestApi
ApiGatewayStage:
  Type: AWS::ApiGateway::Stage
  Properties:
    StageName: Example
    Description: Example Stage
    CacheClusterEnabled: true
    CacheClusterSize: 0.5
    RestApiId:
      Ref: RestApi
    DeploymentId:
      Ref: Deployment
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
RestApi:
  Type: AWS::ApiGateway::RestApi
  Properties:
    Name: ExampleRestApi
GetMethod:
  DependsOn: PutMethod
  Type: AWS::ApiGateway::Method
  Properties:
    HttpMethod: GET
    RestApiId: Ref: RestApi
    ResourceId:
      Fn::GetAtt:
        - "RestApi"
        - "RootResourceId"
    AuthorizationType: NONE
    MethodResponses:
      - StatusCode: "200"
    Integration:
      Type: MOCK
PutMethod:
  Type: AWS::ApiGateway::Method
  Properties:
    HttpMethod: PUT
    RestApiId: Ref: RestApi
    ResourceId:
      Fn::GetAtt:
        - "RestApi"
        - "RootResourceId"
    AuthorizationType: NONE
    MethodResponses:
      - StatusCode: "200"
    Integration:
      Type: MOCK
Deployment:
  DependsOn: GetMethod
  Type: 'AWS::ApiGateway::Deployment'
  Properties:
    RestApiId: Ref: RestApi
    ApiGatewayStage:
      Type: AWS::ApiGateway::Stage
      Properties:
        StageName: Example
        Description: Example Stage
        CacheClusterEnabled: true
        CacheClusterSize: 0.5
        RestApiId:
          Ref: RestApi
        DeploymentId:
          Ref: Deployment
[CT.APIGATEWAY.PR.4] Require an Amazon API Gateway V2 stage to have access logging activated

This control checks whether Amazon API Gateway V2 stages have access logging enabled. Access logging is supported for HTTP and WebSocket APIs.

- **Control objective**: Establish logging and monitoring
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::ApiGatewayV2::Stage
- **AWS CloudFormation guard rule**: [CT.APIGATEWAY.PR.4 rule specification (p. 250)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.APIGATEWAY.PR.4 rule specification (p. 250)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.APIGATEWAY.PR.4 example templates (p. 253)]

**Explanation**

Access logging allows you to log who has called your API and how the caller gained access to the API. You can create your own log group or choose an existing log group that could be managed by API Gateway.

**Remediation for rule failure**

Provide an AccessLogSettings configuration, setting DestinationArn to the ARN of an Amazon CloudWatch log group and Format to a single line log format configuration.

The examples that follow show how to implement this remediation.

**Amazon API Gateway HTTP API Stage - Example**

Amazon API Gateway HTTP API stage configured to send API access logs to Amazon CloudWatch Logs. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "HttpApiStage": {
        "Type": "AWS::ApiGatewayV2::Stage",
        "Properties": {
            "StageName": "SampleStage",
            "Description": "Sample Stage",
            "ApiId": {
                "Ref": "HttpApi"
            },
            "AccessLogSettings": {
                "DestinationArn": 
```
YAML example

HttpApiStage:
  Type: AWS::ApiGatewayV2::Stage
  Properties:
    StageName: SampleStage
    Description: Sample Stage
    ApiId: !Ref 'HttpApi'
    AccessLogSettings:
      DestinationArn: !GetAtt 'LogGroup.Arn'
      Format: '{"requestId":"$context.requestId", "ip": "$context.identity.sourceIp", "user": "$context.identity.user", "requestTime": "$context.requestTime"}'

CT.APIGATEWAY.PR.4 rule specification

# #####################################################################
##       Rule Specification        ##
#####################################################################
#
# Rule Identifier: api_gw_v2_access_logs_enabled_check
#
# Description: This control checks whether Amazon API Gateway V2 stages have access logging enabled. Access logging is supported for HTTP and WebSocket APIs.
#
# Reports on: AWS::ApiGatewayV2::Stage
#
# Evaluates: AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters: None
#
# Scenarios:
#  Scenario: 1
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#    And: The input document does not contain any APIGatewayV2 stage resources
#    Then: SKIP
#  Scenario: 2
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#    And: The input document contains an APIGatewayV2 stage resource
And: 'AccessLogSettings' has not been provided
Then: FAIL
Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document contains an APIGatewayV2 stage resource
And: 'AccessLogSettings' has been provided
And: 'AccessLogSettings.DestinationArn' has not been provided, or has been provided
as an empty string or invalid local reference
And: 'AccessLogSettings.Format' is provided as a non-empty string
Then: FAIL
Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document contains an APIGatewayV2 stage resource
And: 'AccessLogSettings' has been provided
And: 'AccessLogSettings.DestinationArn' has not been provided, or is an empty string
Then: FAIL
Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document contains an APIGatewayV2 stage resource
And: 'AccessLogSettings' has been provided
And: 'AccessLogSettings.DestinationArn' is provided as a non-empty string or valid
local reference
And: 'AccessLogSettings.Format' has not been provided, or is an empty string
Then: PASS

# Constants

let API_GW_V2_STAGE_TYPE = "AWS::ApiGatewayV2::Stage"
let INPUT_DOCUMENT = this

# Assignments

let api_gateway_v2_stages = Resources.*[ Type == %API_GW_V2_STAGE_TYPE ]

# Primary Rules

rule api_gw_v2_access_logs_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%api_gateway_v2_stages not empty {
  check(%api_gateway_v2_stages.Properties)
  %api_gateway_v2_stages not empty {
    [CT.APIGATEWAY.PR.4]: Require an Amazon API Gateway V2 stage to have access logging
    activated
    [FIX]: Provide an 'AccessLogSettings' configuration, setting 'DestinationArn'
    to the ARN of an Amazon CloudWatch log group and 'Format' to a single line log format
    configuration.
  } >>
}

rule api_gw_v2_access_logs_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %API_GW_V2_STAGE_TYPE) {
  check(%INPUT_DOCUMENT.%API_GW_V2_STAGE_TYPE.resourceProperties)
  %API_GW_V2_STAGE_TYPE resourceProperties {
    [CT.APIGATEWAY.PR.4]: Require an Amazon API Gateway V2 stage to have access logging
    activated
    [FIX]: Provide an 'AccessLogSettings' configuration, setting 'DestinationArn'
    to the ARN of an Amazon CloudWatch log group and 'Format' to a single line log format
    configuration.
  }
}
## Parameterized Rules

```plaintext
rule check(api_gateway_v2_stage) {
  %api_gateway_v2_stage {
    # Scenario 2
    AccessLogSettings exists
    AccessLogSettings is_struct

    AccessLogSettings {
      # Scenario 3
      DestinationArn exists
      check_is_string_and_not_empty(DestinationArn) or
      check_local_references(%INPUT_DOCUMENT, DestinationArn, "AWS::Logs::LogGroup")

      # Scenario 4, 5
      Format exists
      check_is_string_and_not_empty(Format)
    }
  }
}

# Utility Rules

## is_cfn_template(doc) {

```
CT.APIGATEWAY.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  LogGroup:
    Type: AWS::Logs::LogGroup
    Properties:
      RetentionInDays: 7
  HttpApi:
    Type: AWS::ApiGatewayV2::Api
    Properties:
      Name: ExampleApi
      ProtocolType: HTTP
  HttpApiStage:
    Type: 'AWS::ApiGatewayV2::Stage'
    Properties:
      StageName: ExampleStage
      Description: Example Stage
      ApiId:
        Ref: HttpApi
      AccessLogSettings:
        DestinationArn:
          Fn::GetAtt:
            - "LogGroup"
            - "Arn"
        Format: >-
          {"requestId":"$context.requestId", "ip": "$context.identity.sourceIp", "user":"$context.identity.user","requestTime":"$context.requestTime"}
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
  HttpApi:
    Type: AWS::ApiGatewayV2::Api
    Properties:
      Name: ExampleApi
      ProtocolType: HTTP
  HttpApiStage:
    Type: 'AWS::ApiGatewayV2::Stage'
    Properties:
      StageName: ExampleStage
      Description: Example Stage
      ApiId:
        Ref: HttpApi
```

[CT.APIGATEWAY.PR.5] Require Amazon API Gateway V2 Websocket and HTTP routes to specify an authorization type

This control checks whether Amazon API Gateway V2 API routes have an authorization type set.

- **Control objective**: Use strong authentication
• **Implementation**: AWS CloudFormation guard rule
• **Control behavior**: Proactive
• **Resource types**: AWS::ApiGatewayV2::Route, AWS::ApiGatewayV2::ApiGatewayManagedOverrides
• **AWS CloudFormation guard rule**: [CT.APIGATEWAY.PR.5 rule specification (p. 255)](#)

**Details and examples**

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.APIGATEWAY.PR.5 rule specification (p. 255)](#)
• For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.APIGATEWAY.PR.5 example templates (p. 258)](#)

**Explanation**

API Gateway supports multiple mechanisms for controlling and managing access to your Websocket or HTTP API. By specifying an authorization type, you can restrict access to your API, to allow only required users or processes.

**Usage considerations**

- This control applies only to routes created by means of the AWS::ApiGatewayV2::Route resource, and to managed overrides that apply to HTTP API routes that are created through quick create.
- This control does not evaluate HTTP API routes imported using the Body or BodyS3Location properties of AWS::ApiGatewayV2::API resources.

**Remediation for rule failure**

For Amazon API Gateway V2 routes, set AuthorizationType to AWS_IAM, JWT or CUSTOM. For Amazon API Gateway V2 managed route overrides with AuthorizationType, set AuthorizationType to AWS_IAM, JWT or CUSTOM.

The examples that follow show how to implement this remediation.

**Amazon API Gateway V2 Route - Example**

Amazon API Gateway V2 route configured with AWS IAM authorization. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ApiGatewayV2Route": {
    "Type": "AWS::ApiGatewayV2::Route",
    "Properties": {
      "ApiId": {
        "Ref": "WebsocketApi"
      },
      "RouteKey": "$connect",
      "AuthorizationType": "AWS_IAM"
    }
  }
}
```
YAML example

```yaml
ApiGatewayV2Route:
  Type: AWS::ApiGatewayV2::Route
  Properties:
    ApiId: !Ref 'WebsocketApi'
    RouteKey: $connect
    AuthorizationType: AWS_IAM
```

The examples that follow show how to implement this remediation.

**Amazon API Gateway V2 Managed Overrides - Example**

Amazon API Gateway V2 managed overrides configured with AWS IAM authorization. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ApiGatewayManagedOverride": {
    "Type": "AWS::ApiGatewayV2::ApiGatewayManagedOverrides",
    "Properties": {
      "ApiId": {
        "Ref": "HttpApi"
      },
      "Route": {
        "AuthorizationType": "AWS_IAM"
      }
    }
  }
}
```

**YAML example**

```yaml
ApiGatewayManagedOverride:
  Type: AWS::ApiGatewayV2::ApiGatewayManagedOverrides
  Properties:
    ApiId: !Ref 'HttpApi'
    Route:
      AuthorizationType: AWS_IAM
```

**CT.APIGATEWAY.PR.5 rule specification**

```bash
# #####################################################################
## Rule Specification     ##
# #####################################################################
# # Rule Identifier:        #
# api_gw_v2_authorization_type_configured_check  #
# # Description:            #
```
# This control checks whether Amazon API Gateway V2 API routes have an authorization type set.
# Reports on:
# AWS::ApiGatewayV2::Route, AWS::ApiGatewayV2::ApiGatewayManagedOverrides
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain any Amazon API Gateway V2 route or managed route overrides resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Amazon API Gateway V2 managed route overrides resource
# And: In 'Route', 'AuthorizationType' has not been provided
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Amazon API Gateway V2 route resource
# And: 'AuthorizationType' has not been provided
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Amazon API Gateway V2 route or managed route overrides resource
# And: 'AuthorizationType' has been provided and set to a value other than 'AWS_IAM', 'JWT' or 'CUSTOM'
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Amazon API Gateway V2 route or managed route overrides resource
# And: 'AuthorizationType' has been provided and set to a value of 'AWS_IAM', 'JWT' or 'CUSTOM'
# Then: PASS

# Constants

let API_GW_ROUTE_TYPE = "AWS::ApiGatewayV2::Route"
let API_GW_MANAGED_OVERRIDE_TYPE = "AWS::ApiGatewayV2::ApiGatewayManagedOverrides"
let ALLOWED_AUTHORIZATION_TYPES = ["AWS_IAM", "JWT", "CUSTOM"]
let INPUT_DOCUMENT = this

# Assignments

let api_route = Resources.*[ Type == %API_GW_ROUTE_TYPE ]
let api_override = Resources.*[ Type == %API_GW_MANAGED_OVERRIDE_TYPE ]
rule api_gw_v2_authorization_type_configured_check when is_cfn_template(%INPUT_DOCUMENT) %api_route not empty {
    check_api_route(%api_route.Properties)
    <<
    [CT.APIGATEWAY.PR.5]: Require Amazon API Gateway V2 Websocket and HTTP routes to specify an authorization type
    [FIX]: For Amazon API Gateway V2 routes, set 'AuthorizationType' to 'AWS_IAM', 'JWT' or 'CUSTOM'. For Amazon API Gateway V2 managed route overrides with 'AuthorizationType', set 'AuthorizationType' to 'AWS_IAM', 'JWT' or 'CUSTOM'.
    >>
}

rule api_gw_v2_authorization_type_configured_check when is_cfn_template(%INPUT_DOCUMENT) %api_override not empty {
    check_api_override(%api_override.Properties)
    <<
    [CT.APIGATEWAY.PR.5]: Require Amazon API Gateway V2 Websocket and HTTP routes to specify an authorization type
    [FIX]: For Amazon API Gateway V2 routes, set 'AuthorizationType' to 'AWS_IAM', 'JWT' or 'CUSTOM'. For Amazon API Gateway V2 managed route overrides with 'AuthorizationType', set 'AuthorizationType' to 'AWS_IAM', 'JWT' or 'CUSTOM'.
    >>
}

rule api_gw_v2_authorization_type_configured_check when is_cfn_hook(%INPUT_DOCUMENT, %API_GW_ROUTE_TYPE) {
    check_api_route(%INPUT_DOCUMENT.%API_GW_ROUTE_TYPE.resourceProperties)
    <<
    [CT.APIGATEWAY.PR.5]: Require Amazon API Gateway V2 Websocket and HTTP routes to specify an authorization type
    [FIX]: For Amazon API Gateway V2 routes, set 'AuthorizationType' to 'AWS_IAM', 'JWT' or 'CUSTOM'. For Amazon API Gateway V2 managed route overrides with 'AuthorizationType', set 'AuthorizationType' to 'AWS_IAM', 'JWT' or 'CUSTOM'.
    >>
}

rule api_gw_v2_authorization_type_configured_check when is_cfn_hook(%INPUT_DOCUMENT, %API_GW_MANAGED_OVERRIDE_TYPE) {
    check_api_override(%INPUT_DOCUMENT.%API_GW_MANAGED_OVERRIDE_TYPE.resourceProperties)
    <<
    [CT.APIGATEWAY.PR.5]: Require Amazon API Gateway V2 Websocket and HTTP routes to specify an authorization type
    [FIX]: For Amazon API Gateway V2 routes, set 'AuthorizationType' to 'AWS_IAM', 'JWT' or 'CUSTOM'. For Amazon API Gateway V2 managed route overrides with 'AuthorizationType', set 'AuthorizationType' to 'AWS_IAM', 'JWT' or 'CUSTOM'.
    >>
}

# Parameterized Rules
#
rule check_api_route(api_route) {
    %api_route {
        # Scenario 3
        # Scenario 4 and 5
        AuthorizationType exists
        AuthorizationType in %ALLOWED_AUTHORIZATION_TYPES
    }
}

rule check_api_override(api_override) {
    %api_override [
        # Scenario 2
        Route exists
        Route is_struct
    ]
CT.APIGATEWAY.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  ApiGatewayV2Route:
    Type: AWS::ApiGatewayV2::Route
    Properties:
      ApiId: a1bcdef2gh
      RouteKey: $connect
      AuthorizationType: AWS_IAM
```

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  ApiGatewayManagedOverride:
    Type: AWS::ApiGatewayV2::ApiGatewayManagedOverrides
    Properties:
      ApiId: a1bcdef2gh
      Route:
        AuthorizationType: AWS_IAM
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```yaml
Resources:
  ApiGatewayV2Route:
    Type: AWS::ApiGatewayV2::Route
    Properties:
      ApiId: a1bcdef2gh
```
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```json
Resources:
  ApiGatewayManagedOverride:
    Type: AWS::ApiGatewayV2::ApiGatewayManagedOverrides
    Properties:
      ApiId: a1bcdef2gh
      Route:
        AuthorizationType: NONE
```

AWS Certificate Manager controls

Topics
- [CT.ACM.PR.1] Require an AWS Private CA certificate to have a single domain name (p. 259)

[CT.ACM.PR.1] Require an AWS Private CA certificate to have a single domain name

This control checks whether any AWS Certificate Manager (ACM) Private CA certificates have wildcard domain names instead of single domain names.

- **Control objective**: Protect configurations
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::CertificateManager::Certificate
- **AWS CloudFormation guard rule**: [CT.ACM.PR.1 rule specification (p. 261)]

Details and examples
- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ACM.PR.1 rule specification (p. 261)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ACM.PR.1 example templates (p. 263)]

Explanation

AWS Private CA allows you to use wildcards (*) in the domain name, so you can protect several sites in the same domain. This type of certificate presents some risk, because if the private key of a certificate is compromised, all domain and subdomains with the compromised certificate are compromised. We recommend that you use single domain name certificates instead of wildcard certificates to reduce these associated risks.

Remediation for rule failure

Set DomainName and each entry within SubjectAlternativeNames to a fully qualified domain name (FQDN) that does not contain a wildcard (*).
The examples that follow show how to implement this remediation.

**AWS Certificate Manager Private CA Certificate - Example One**

AWS Certificate Manager Private CA certificate configured with a single domain and no subject alternative names. The example is shown in JSON and in YAML.

**JSON example**

```
{
  "Resources": {
    "ACMCertificate": {
      "Type": "AWS::CertificateManager::Certificate",
      "Properties": {
        "DomainName": "example.com"
      }
    }
  }
}
```

**YAML example**

```
Resources:
  ACMCertificate:
    Type: AWS::CertificateManager::Certificate
    Properties:
      DomainName: example.com
```

The examples that follow show how to implement this remediation.

**AWS Certificate Manager Private CA Certificate - Example Two**

AWS Certificate Manager private CA certificate configured with a single domain and one subject alternative name. The example is shown in JSON and in YAML.

**JSON example**

```
{
  "Resources": {
    "ACMCertificate": {
      "Type": "AWS::CertificateManager::Certificate",
      "Properties": {
        "DomainName": "example.com",
        "SubjectAlternativeNames": [
          "www.example.com"
        ]
      }
    }
  }
}
```
YAML example

Resources:
  ACMCertificate:
    Type: AWS::CertificateManager::Certificate
    Properties:
      DomainName: example.com
      SubjectAlternativeNames:
        - www.example.com

CT.ACM.PR.1 rule specification

# #######################################################################
##       Rule Specification       ##
#######################################################################
#
# Rule Identifier:
#   acm_certificate_domain_name_check
#
# Description:
#   This control checks whether any AWS Certificate Manager (ACM) Private CA certificates have wildcard domain names instead of single domain names.
#
# Reports on:
#   AWS::CertificateManager::Certificate
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any ACM certificate resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ACM certificate resource
#     And: 'CertificateAuthorityArn' has not been provided
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ACM certificate resource
#     And: 'CertificateAuthorityArn' has been provided
#     And: 'DomainName' has not been provided
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document

And: The input document contains an ACM certificate resource
And: 'CertificateAuthorityArn' has been provided
And: 'SubjectAlternativeNames' has not been provided or provided as an empty list
And: 'DomainName' has been provided with a string that begins with a wildcard character ('*').
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ACM certificate resource
And: 'CertificateAuthorityArn' has been provided
And: 'DomainName' has been provided with a string that does not begin with a wildcard character ('*').
And: 'SubjectAlternativeNames' has been provided as a non-empty list containing a string that begins with a wildcard character ('*').
Then: FAIL

Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ACM certificate resource
And: 'CertificateAuthorityArn' has been provided
And: 'DomainName' has been provided with a string that does not begin with a wildcard character ('*').
And: 'SubjectAlternativeNames' has been provided as a non-empty list containing a string that begins with a wildcard character ('*').
Then: PASS

Scenario: 7
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ACM certificate resource
And: 'CertificateAuthorityArn' has been provided
And: 'DomainName' has been provided with a string that does not begin with a wildcard character ('*').
And: 'SubjectAlternativeNames' has been provided as a non-empty list where no entries are strings that begin with a wildcard character ('*').
Then: PASS

Constants

let ACM_CERTIFICATE_TYPE = "AWS::CertificateManager::Certificate"
let WILDCARD_DOMAIN_NAME_REGEX_PATTERN = /^\(^*\).*$/
let INPUT_DOCUMENT = this

Assignments

let acm_certificates = Resources.*[ Type == %ACM_CERTIFICATE_TYPE ]

Primary Rules

rule acm_certificate_domain_name_check when is_cfn_template(%INPUT_DOCUMENT)
%acm_certificates not empty {
check(%acm_certificates.Properties)
  [CT.ACM.PR.1]: Require an AWS Private CA certificate to have a single domain name
  [FIX]: Set 'DomainName' and each entry within 'SubjectAlternativeNames' to a fully qualified domain name (FQDN) that does not contain a wildcard (*).
}

rule acm_certificate_domain_name_check when is_cfn_hook(%INPUT_DOCUMENT, %ACM_CERTIFICATE_TYPE) {
  check(%INPUT_DOCUMENT.%ACM_CERTIFICATE_TYPE.resourceProperties)
<<
[CT.ACM.PR.1]: Require an AWS Private CA certificate to have a single domain name
[Fix]: Set 'DomainName' and each entry within 'SubjectAlternativeNames' to a
fully qualified domain name (FQDN) that does not contain a wildcard (*).
>>
#
# Parameterized Rules
#
rule check(acm_certificate) {
  %acm_certificate [
    CertificateAuthorityArn exists
  ] {
    # Scenario 2
    DomainName exists
    # Scenario 3 and 4
    check_wildcarded_domain(DomainName)
    check_subject_alternative_names(this)
  }
}
rule check_subject_alternative_names(acm_certificate) {
  %acm_certificate [ 
    SubjectAlternativeNames exists
    SubjectAlternativeNames is_list
    SubjectAlternativeNames not empty
  ] {
    SubjectAlternativeNames[*] {
      check_wildcarded_domain(this)
    }
  }
}
rule check_wildcarded_domain(domain) {
  %domain {
    this is_string
    this != %WILDCARD_DOMAIN_NAME_REGEX_PATTERN
  }
}
#
# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.ACM.PR.1 example templates
You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.
PASS Example - Use this template to verify a compliant resource creation.
AWS AppSync controls

Topics
- [CT.APPSYNC.PR.1] Require an AWS AppSync GraphQL API to have logging enabled (p. 264)

[CT.APPSYNC.PR.1] Require an AWS AppSync GraphQL API to have logging enabled

This control checks whether an AWS AppSync GraphQL API has been configured to send request-level and field-level logs to Amazon CloudWatch Logs.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::AppSync::GraphQLApi
- **AWS CloudFormation guard rule:** [CT.APPSYNC.PR.1 rule specification](p. 265)

Details and examples
- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.APPSYNC.PR.1 rule specification](p. 265)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.APPSYNC.PR.1 example templates](p. 268)

Explanation

AppSync logs are useful for debugging issues related to requests.

Remediation for rule failure

Within LogConfig, set FieldLogLevel to ALL or ERROR and set CloudWatchLogsRoleArn to the ARN of an AWS IAM role configured to allow AWS AppSync to send logs to Amazon CloudWatch Logs.
The examples that follow show how to implement this remediation.

**AWS AppSync GraphQL API - Example**

An AWS AppSync GraphQL API configured to send GraphQL operations and tracing to Amazon CloudWatch Logs. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "GraphQLApi": {
        "Type": "AWS::AppSync::GraphQLApi",
        "Properties": {
            "Name": "SampleApi",
            "AuthenticationType": "AWS_IAM",
            "LogConfig": {
                "FieldLogLevel": "ALL",
                "CloudWatchLogsRoleArn": {
                    "Fn::GetAtt": [
                        "AppSyncLoggingRole",
                        "Arn"
                    ]
                }
            }
        }
    }
}
```

**YAML example**

```yaml
GraphQLApi:
  Type: AWS::AppSync::GraphQLApi
  Properties:
    Name: SampleApi
    AuthenticationType: AWS_IAM
    LogConfig:
      FieldLogLevel: ALL
      CloudWatchLogsRoleArn: !GetAtt 'AppSyncLoggingRole.Arn'
```

**CT.APPSYNC.PR.1 rule specification**

```text
# ##################################################
##       Rule Specification        ##
# ##################################################

# Rule Identifier:  
# appsync_logging_enabled_check

# Description:  
# This control checks whether an AWS AppSync GraphQL API has been configured to send request-level and field-level logs to Amazon CloudWatch Logs.

# Reports on:  
# AWS::AppSync::GraphQLApi

# Evaluates:  
# AWS CloudFormation, AWS CloudFormation hook
```

265
# Proactive controls

## Rule Parameters:
None

## Scenarios:

### Scenario: 1
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document does not contain any AppSync GraphQL API resources
Then: SKIP

### Scenario: 2
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an AppSync GraphQL API resource
And: 'LogConfig' has not been provided
Then: FAIL

### Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an AppSync GraphQL API resource
And: 'LogConfig' has been provided
And: 'FieldLogLevel' in 'LogConfig' has not been provided or provided and set to a value other than 'ERROR' or 'ALL'
And: 'CloudWatchLogsRoleArn' in 'LogConfig' has not been provided or provided and set to an empty string or invalid local reference
Then: FAIL

### Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an AppSync GraphQL API resource
And: 'LogConfig' has been provided
And: 'FieldLogLevel' in 'LogConfig' has been provided and set to 'ERROR' or 'ALL'
And: 'CloudWatchLogsRoleArn' in 'LogConfig' has not been provided or provided and set to an empty string or invalid local reference
Then: FAIL

### Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an AppSync GraphQL API resource
And: 'LogConfig' has been provided
And: 'FieldLogLevel' in 'LogConfig' has not been provided or provided and set to a non-empty string or valid local reference
Then: FAIL

### Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an AppSync GraphQL API resource
And: 'LogConfig' has been provided
And: 'FieldLogLevel' in 'LogConfig' has been provided and set to 'ERROR' or 'ALL'
And: 'CloudWatchLogsRoleArn' in 'LogConfig' has been provided and set to a non-empty string or valid local reference
Then: PASS

## Constants

```javascript
let APPSYNC_GRAPHQL_API_TYPE = "AWS::AppSync::GraphQLApi"
let ALLOWED_APPSYNC_LOG_LEVELS = ["ERROR", "ALL"]
```
let INPUT_DOCUMENT = this

# # Assignments
# let appsync_graphql_apis = Resources.*[ Type == %APPSYNC_GRAPHQL_API_TYPE ]

# # Primary Rules
# rule appsync_logging_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
#   %appsync_graphql_apis not empty {
#     check(%appsync_graphql_apis.Properties)
#     <<<
#     [CT.APPSYNC.PR.1]: Require an AWS AppSync GraphQL API to have logging enabled
#     [FIX]: Within 'LogConfig', set 'FieldLogLevel' to 'ALL' or 'ERROR' and set
#     'CloudWatchLogsRoleArn' to the ARN of an AWS IAM role configured to allow AWS AppSync to
#     send logs to Amazon CloudWatch Logs.
#     >>>
#   }
#   rule appsync_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %APPSYNC_GRAPHQL_API_TYPE) {
#     check(%INPUT_DOCUMENT.%APPSYNC_GRAPHQL_API_TYPE.resourceProperties)
#     <<<
#     [CT.APPSYNC.PR.1]: Require an AWS AppSync GraphQL API to have logging enabled
#     [FIX]: Within 'LogConfig', set 'FieldLogLevel' to 'ALL' or 'ERROR' and set
#     'CloudWatchLogsRoleArn' to the ARN of an AWS IAM role configured to allow AWS AppSync to
#     send logs to Amazon CloudWatch Logs.
#     >>>
#   }
#
# # Parameterized Rules
# rule check(appsync_graphql_api) {
#   %appsync_graphql_api {
#     # Scenario 2
#     LogConfig exists
#     LogConfig is_struct
#     LogConfig {
#       # Scenarios 3, 4, 5 and 6
#       FieldLogLevel exists
#       FieldLogLevel in %ALLOWED_APPSYNC_LOG_LEVELS
#       CloudWatchLogsRoleArn exists
#       check_is_string_and_not_empty(CloudWatchLogsRoleArn) or
#       check_local_references(%INPUT_DOCUMENT, CloudWatchLogsRoleArn, "AWS::IAM::Role")
#     }
#   }
# }
#
# # Utility Rules
# rule is_cfn_template(doc) {
#   %doc {
#     AWSTemplateFormatVersion exists or
#     Resources exists
#   }
# }
# rule is_cfn_hook(doc, RESOURCE_TYPE) {
#   %doc.%RESOURCE_TYPE.resourceProperties exists

CT.APPSYNC.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
GraphQLApi:
  Type: AWS::AppSync::GraphQLApi
  Properties:
    Name:
      Fn::Sub: ${AWS::StackName}-example
    AuthenticationType: AWS_IAM
    LogConfig:
      FieldLogLevel: ALL
      CloudWatchLogsRoleArn:
        Fn::GetAtt:
        - AppSyncLoggingRole
        - Arn
    AppSyncLoggingRole:
      Type: AWS::IAM::Role
      Properties:
        AssumeRolePolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Principal:
                Service:
                  - appsync.amazonaws.com
              Action:
                - sts:AssumeRole
              Path: /
AWS Control Tower User Guide
Proactive controls

Policies:
- PolicyName: AppSyncLoggingPolicy
  PolicyDocument:
    Version: '2012-10-17'
    Statement:
      - Effect: Allow
        Action:
        - logs:CreateLogGroup
        - logs:CreateLogStream
        - logs:PutLogEvents
        Resource: '*'

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
GraphQLApi:
  Type: AWS::AppSync::GraphQLApi
  Properties:
    Name:
      Fn::Sub: ${AWS::StackName}-example
    AuthenticationType: AWS_IAM
    LogConfig:
      FieldLogLevel: NONE
      CloudWatchLogsRoleArn:
        Fn::GetAtt:
          - AppSyncLoggingRole
          - Arn
    AppSyncLoggingRole:
      Type: AWS::IAM::Role
      Properties:
        AssumeRolePolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Principal:
                Service:
                  - appsync.amazonaws.com
              Action:
                - sts:AssumeRole
              Path: /
          Policies:
            - PolicyName: AppSyncLoggingPolicy
              PolicyDocument:
                Version: '2012-10-17'
                Statement:
                  - Effect: Allow
                    Action:
                    - logs:CreateLogGroup
                    - logs:CreateLogStream
                    - logs:PutLogEvents
                    Resource: '*'

Amazon Athena controls

Topics
- [CT.ATHENA.PR.1] Require an Amazon Athena workgroup to encrypt Athena query results at rest (p. 270)
[CT.ATHENA.PR.2] Require an Amazon Athena workgroup to encrypt Athena query results at rest with an AWS Key Management Service (KMS) key (p. 274)

[CT.ATHENA.PR.1] Require an Amazon Athena workgroup to encrypt Athena query results at rest

This control checks whether an Amazon Athena workgroup requires query results to be encrypted at rest.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Athena::WorkGroup
- **AWS CloudFormation guard rule:** [CT.ATHENA.PR.1 rule specification (p. 271)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ATHENA.PR.1 rule specification (p. 271)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ATHENA.PR.1 example templates (p. 273)]

**Explanation**

For an added layer of security, you can encrypt the results of all Athena queries in Amazon S3. The location where Athena stores these query results is known as the Amazon S3 results location.

**Usage considerations**

- This control requires Athena workgroups to override client settings by requiring the EnforceWorkGroupConfiguration property to be provided and set to true, or omitted to adopt the default value of true.

**Remediation for rule failure**

In the WorkGroupConfiguration.ResultConfiguration parameter, provide an EncryptionConfiguration configuration with an EncryptionOption value set to one of CSE_KMS, SSE_KMS or SSE_S3.

The examples that follow show how to implement this remediation.

**Amazon Athena workgroup - Example**

Amazon Athena workgroup configured to encrypt Athena query results with Amazon S3-managed keys (SSE_S3). The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "AthenaWorkGroup": {
      "Type": "AWS::Athena::WorkGroup",
      "Properties": {
         "Name": {
            "Fn::Sub": "${AWS::StackName}-example"
         },
         "Description": "Example workgroup",
         "State": "ENABLED",
         "EncryptionConfiguration": {
            "EncryptionOption": "SSE_S3",
            "KMSEncryptionKeyArn": "arn:aws:kms:us-east-1:123456789012:key/012345678901234567"  
         }
      }
   }
}
```
"WorkGroupConfiguration": {
  "ResultConfiguration": {
    "EncryptionConfiguration": {
      "EncryptionOption": "SSE_S3"
    }
  }
}

YAML example

AthenaWorkGroup:
  Type: AWS::Athena::WorkGroup
  Properties:
    Name: !Sub '${AWS::StackName}-example'
    Description: Example workgroup
    State: ENABLED
    WorkGroupConfiguration:
      ResultConfiguration:
        EncryptionConfiguration:
          EncryptionOption: SSE_S3

CT.ATHENA.PR.1 rule specification

# #########################################################################
##       Rule Specification        ##
#########################################################################

# Rule Identifier:
#   athena_workgroup_results_encrypted_at_rest_check
#
# Description:
#   This control checks whether an Amazon Athena workgroup requires query results to be
#   encrypted at rest.
#
# Reports on:
#   AWS::Athena::WorkGroup
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document does not contain any Athena workgroup resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document contains an Athena workgroup resource
#     And: 'EnforceWorkGroupConfiguration' in 'WorkGroupConfiguration' has been provided
#
# Constants
#
let ATHENA_WORKGROUP_TYPE = "AWS::Athena::WorkGroup"
let INPUT_DOCUMENT = this

# Assignments
#
let athena_workgroups = Resources.*[ Type == %ATHENA_WORKGROUP_TYPE ]

# Primary Rules
#
rule athena_workgroup_results_encrypted_at_rest_check when is_cfn_template(%INPUT_DOCUMENT) {%
    check(%athena_workgroups.Properties)
    %athena_workgroups not empty {
        <<
            [CT.ATHENA.PR.1]: Require an Amazon Athena workgroup to encrypt Athena query results at rest
            [FIX]: In the 'WorkGroupConfiguration.ResultConfiguration' parameter, provide an 'EncryptionConfiguration' configuration with an 'EncryptionOption' value set to one of 'CSE_KMS', 'SSE_KMS' or 'SSE_S3'.
        >>
    }
}

rule athena_workgroup_results_encrypted_at_rest_check when is_cfn_hook(%INPUT_DOCUMENT,%ATHENA_WORKGROUP_TYPE) {%
    check(%INPUT_DOCUMENT.%ATHENA_WORKGROUP_TYPE.resourceProperties)

    # Scenario: 3
    # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
    # And: The input document contains an Athena workgroup resource
    # And: 'EnforceWorkGroupConfiguration' in 'WorkGroupConfiguration' has not been provided or provided and set to bool(true)
    # And: 'EncryptionConfiguration' in 'WorkGroupConfiguration.ResultConfiguration' has not been provided
    # Then: FAIL

    # Scenario: 4
    # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
    # And: The input document contains an Athena workgroup resource
    # And: 'EnforceWorkGroupConfiguration' in 'WorkGroupConfiguration' has not been provided or provided and set to bool(true)
    # And: 'EncryptionConfiguration' in 'WorkGroupConfiguration.ResultConfiguration' has been provided
    # And: 'EncryptionOption' in 'EncryptionConfiguration' has not been provided or provided as an empty string
    # Then: FAIL

    # Scenario: 5
    # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
    # And: The input document contains an Athena workgroup resource
    # And: 'EnforceWorkGroupConfiguration' in 'WorkGroupConfiguration' has not been provided or provided and set to bool(true)
    # And: 'EncryptionConfiguration' in 'WorkGroupConfiguration.ResultConfiguration' has been provided
    # And: 'EncryptionOption' in 'EncryptionConfiguration' has been provided as a non-empty string
    # Then: PASS

    # Constants
    #
    let ATHENA_WORKGROUP_TYPE = "AWS::Athena::WorkGroup"
    let INPUT_DOCUMENT = this

    # Assignments
    #
    let athena_workgroups = Resources.*[ Type == %ATHENA_WORKGROUP_TYPE ]

    # Primary Rules
    #
    rule athena_workgroup_results_encrypted_at_rest_check when is_cfn_template(%INPUT_DOCUMENT) {
        check(%athena_workgroups.Properties)
        %athena_workgroups not empty {
            <<
                [CT.ATHENA.PR.1]: Require an Amazon Athena workgroup to encrypt Athena query results at rest
                [FIX]: In the 'WorkGroupConfiguration.ResultConfiguration' parameter, provide an 'EncryptionConfiguration' configuration with an 'EncryptionOption' value set to one of 'CSE_KMS', 'SSE_KMS' or 'SSE_S3'.
            >>
        }
    }
    rule athena_workgroup_results_encrypted_at_rest_check when is_cfn_hook(%INPUT_DOCUMENT,%ATHENA_WORKGROUP_TYPE) {
        check(%INPUT_DOCUMENT.%ATHENA_WORKGROUP_TYPE.resourceProperties)
[CT.ATHENA.PR.1]: Require an Amazon Athena workgroup to encrypt Athena query results at rest

[FIX]: In the 'WorkGroupConfiguration.ResultConfiguration' parameter, provide an 'EncryptionConfiguration' configuration with an 'EncryptionOption' value set to one of 'CSE_KMS', 'SSE_KMS' or 'SSE_S3'.

# Parameterized Rules

## CT.ATHENA.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.
PASS Example - Use this template to verify a compliant resource creation.

Resources:
AthenaWorkGroup:
  Type: AWS::Athena::WorkGroup
  Properties:
    Name:
      Fn::Sub: ${{AWS::StackName}}-example
    Description: Example workgroup
    State: ENABLED
    WorkGroupConfiguration:
      ResultConfiguration:
        EncryptionConfiguration:
          EncryptionOption: SSE_S3

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
AthenaWorkGroup:
  Type: AWS::Athena::WorkGroup
  Properties:
    Name:
      Fn::Sub: ${{AWS::StackName}}-example
    Description: Example workgroup
    State: ENABLED
    WorkGroupConfiguration:
      EnforceWorkGroupConfiguration: false

[CT.ATHENA.PR.2] Require an Amazon Athena workgroup to encrypt Athena query results at rest with an AWS Key Management Service (KMS) key

This control checks whether an Amazon Athena workgroup is configured to encrypt query results at rest with an AWS KMS key.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Athena::WorkGroup
- **AWS CloudFormation guard rule:** [CT.ATHENA.PR.2 rule specification (p. 276)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ATHENA.PR.2 rule specification (p. 276)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ATHENA.PR.2 example templates (p. 279)]

Explanation

For an added layer of security, you can encrypt the results of Athena queries in the workgroup with AWS Key Management Service (KMS).
Usage considerations

- This control requires an Athena workgroup to override client settings by requiring the EnforceWorkGroupConfiguration property to be provided and set to true, or omitted to adopt the default value of true.

Remediation for rule failure

In the WorkGroupConfiguration.ResultConfiguration parameter, provide an EncryptionConfiguration configuration with an EncryptionOption set to a KMS-based encryption option, and with KmsKey set to the identifier or ARN of an AWS KMS key, or the name of an AWS KMS key alias.

The examples that follow show how to implement this remediation.

Amazon Athena workgroup - Example

Amazon Athena workgroup configured to encrypt Athena query results with AWS KMS (SSE_KMS). The example is shown in JSON and in YAML.

JSON example

```json
{
   "AthenaWorkGroup": {
      "Type": "AWS::Athena::WorkGroup",
      "Properties": {
         "Name": {
            "Fn::Sub": "${AWS::StackName}-example"
         },
         "Description": "Example workgroup",
         "State": "ENABLED",
         "WorkGroupConfiguration": {
            "EnforceWorkGroupConfiguration": true,
            "ResultConfiguration": {
               "EncryptionConfiguration": {
                  "KmsKey": {
                     "Ref": "Key"
                  },
                  "EncryptionOption": "SSE_KMS"
               }
            }
         }
      }
   }
}
```

YAML example

```
AthenaWorkGroup:
   Type: AWS::Athena::WorkGroup
   Properties:
      Name: !Sub '${AWS::StackName}-example'
      Description: Example workgroup
      State: ENABLED
      WorkGroupConfiguration:
         EnforceWorkGroupConfiguration: true
         ResultConfiguration:
            EncryptionConfiguration:
               KmsKey: !Ref 'Key'
```
EncryptionOption: SSE_KMS

CT.ATHENAPR.2 rule specification

```plaintext
# ###################################################################
##       Rule Specification        ##
###################################################################
#
# Rule Identifier:
#   athena_workgroup_results_encrypted_at_rest_kms_check
#
# Description:
#   This control checks whether an Amazon Athena workgroup is configured to encrypt query results at rest with an AWS KMS key.
#
# Reports on:
#   AWS::Athena::WorkGroup
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any Athena workgroup resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Athena workgroup resource
#     And: 'EnforceWorkGroupConfiguration' in 'WorkGroupConfiguration' has been provided and set to a value other than bool(true)
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Athena workgroup resource
#     And: 'EnforceWorkGroupConfiguration' in 'WorkGroupConfiguration' has not been provided or provided and set to bool(true)
#     And: 'EncryptionConfiguration' in 'WorkGroupConfiguration.ResultConfiguration' has not been provided
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Athena workgroup resource
#     And: 'EnforceWorkGroupConfiguration' in 'WorkGroupConfiguration' has not been provided or provided and set to bool(true)
#     And: 'EncryptionConfiguration' in 'WorkGroupConfiguration.ResultConfiguration' has been provided
#     And: 'EncryptionOption' in 'EncryptionConfiguration' has not been provided or provided as an empty string
#     And: 'KmsKey' in 'EncryptionConfiguration' has not been provided or provided as an empty string or
```
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Athena workgroup resource
# And: 'EnforceWorkGroupConfiguration' in 'WorkGroupConfiguration' has not been provided or provided
# and set to bool(true)
# And: 'EncryptionConfiguration' in 'WorkGroupConfiguration.ResultConfiguration' has been provided
# And: 'EncryptionOption' in 'EncryptionConfiguration' has been provided as a non-empty string
# And: 'KmsKey' in 'EncryptionConfiguration' has not been provided or provided as an empty string or
# invalid local reference
# Then: FAIL

# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Athena workgroup resource
# And: 'EnforceWorkGroupConfiguration' in 'WorkGroupConfiguration' has not been provided or provided
# and set to bool(true)
# And: 'EncryptionConfiguration' in 'WorkGroupConfiguration.ResultConfiguration' has been provided
# And: 'EncryptionOption' in 'EncryptionConfiguration' has not been provided or provided as an empty string
# And: 'KmsKey' in 'EncryptionConfiguration' has been provided as a non-empty string or valid local reference to
# a KMS key or key alias
# Then: FAIL

# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Athena workgroup resource
# And: 'EnforceWorkGroupConfiguration' in 'WorkGroupConfiguration' has not been provided or provided
# and set to bool(true)
# And: 'EncryptionConfiguration' in 'WorkGroupConfiguration.ResultConfiguration' has been provided
# And: 'EncryptionOption' in 'EncryptionConfiguration' has been provided as a non-empty string
# And: 'KmsKey' in 'EncryptionConfiguration' has been provided as a non-empty string or valid local reference to
# a KMS key or key alias
# Then: PASS

# Constants
let ATHENA_WORKGROUP_TYPE = "AWS::Athena::WorkGroup"
let INPUT_DOCUMENT = this

# Assignments
let athena_workgroups = Resources.*[ Type == %ATHENA_WORKGROUP_TYPE ]

# Primary Rules
rule athena_workgroup_results_encrypted_at_rest_kms_check when
is_cfn_template(INPUT_DOCUMENT)
%athena_workgroups not empty
{


check(%athena_workgroups.Properties)
<<
[CT.ATHENA.PR.2]: Require an Amazon Athena workgroup to encrypt Athena query results at rest with an AWS Key Management Service (KMS) key
[FIX]: In the 'WorkGroupConfiguration.ResultConfiguration' parameter, provide an 'EncryptionConfiguration' configuration with an 'EncryptionOption' set to a KMS-based encryption option, and with 'KmsKey' set to the identifier or ARN of an AWS KMS key, or the name of an AWS KMS key alias.
>>

rule athena_workgroup_results_encrypted_at_rest_kms_check when is_cfn_hook(%INPUT_DOCUMENT, %ATHENA_WORKGROUP_TYPE) {
    check(%INPUT_DOCUMENT.%ATHENA_WORKGROUP_TYPE.resourceProperties)
    <<
    [CT.ATHENA.PR.2]: Require an Amazon Athena workgroup to encrypt Athena query results at rest with an AWS Key Management Service (KMS) key
    [FIX]: In the 'WorkGroupConfiguration.ResultConfiguration' parameter, provide an 'EncryptionConfiguration' configuration with an 'EncryptionOption' set to a KMS-based encryption option, and with 'KmsKey' set to the identifier or ARN of an AWS KMS key, or the name of an AWS KMS key alias.
    >>
}

# Parameterized Rules
#
rule check(athena_workgroup) {
    %athena_workgroup {  
        WorkGroupConfiguration exists
        WorkGroupConfiguration is_struct
        WorkGroupConfiguration {
            # Scenario 2
            EnforceWorkGroupConfiguration not exists or
            EnforceWorkGroupConfiguration == true

            ResultConfiguration exists
            ResultConfiguration is_struct
            ResultConfiguration {
                # Scenario 3
                EncryptionConfiguration exists
                EncryptionConfiguration is_struct

                EncryptionConfiguration {
                    # Scenarios 4, 5, 6 and 7
                    EncryptionOption exists
                    check_is_string_and_not_empty(EncryptionOption)

                    KmsKey exists
                    check_is_string_and_not_empty(KmsKey) or
                    check_local_references(%INPUT_DOCUMENT, KmsKey, "AWS::KMS::Key") or
                    check_local_references(%INPUT_DOCUMENT, KmsKey, "AWS::KMS::Alias")
                }
            }
        }
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  Key:
    Type: AWS::KMS::Key
    Properties:
      Version: 2012-10-17
      Id: example-policy
      Statement:
        - Sid: Enable IAM user permissions
          Effect: Allow
          Principal:
            AWS:
              Fn::Sub: arn:${AWS::Partition}:iam::${AWS::AccountId}:root
          Action: kms:*
          Resource: '*'
     KeySpec: SYMMETRIC_DEFAULT

AthenaWorkGroup:
  Type: AWS::Athena::WorkGroup
  Properties:
    Name:
      Fn::Sub: ${AWS::StackName}-example
Description: Example workgroup
State: ENABLED
WorkGroupConfiguration:
    EnforceWorkGroupConfiguration: true
ResultConfiguration:
    EncryptionConfiguration:
        KmsKey:
            Ref: Key
        EncryptionOption: SSE_KMS

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
AthenaWorkGroup:
    Type: AWS::Athena::WorkGroup
    Properties:
        Name:
            Fn::Sub: ${AWS::StackName}-example
        Description: Example workgroup
        State: ENABLED
        WorkGroupConfiguration:
            ResultConfiguration:
                EncryptionConfiguration:
                    EncryptionOption: SSE_S3

Amazon CloudFront controls

Topics
- [CT.CLOUDFRONT.PR.1] Require an Amazon CloudFront distribution to have a default root object configured (p. 281)
- [CT.CLOUDFRONT.PR.2] Require any Amazon CloudFront distributions with Amazon S3 backed origins to have an origin access identity configured (p. 285)
- [CT.CLOUDFRONT.PR.3] Require an Amazon CloudFront distribution to have encryption in transit configured (p. 293)
- [CT.CLOUDFRONT.PR.4] Require an Amazon CloudFront distribution to have origin failover configured (p. 299)
- [CT.CLOUDFRONT.PR.5] Require any Amazon CloudFront distribution to have logging enabled (p. 305)
- [CT.CLOUDFRONT.PR.6] Require an Amazon CloudFront distribution to use custom SSL/TLS certificates (p. 311)
- [CT.CLOUDFRONT.PR.7] Require an Amazon CloudFront distribution to use SNI to serve HTTPS requests (p. 317)
- [CT.CLOUDFRONT.PR.8] Require an Amazon CloudFront distribution to encrypt traffic to custom origins (p. 324)
- [CT.CLOUDFRONT.PR.9] Require an Amazon CloudFront distribution to have a security policy of TLSv1.2 as a minimum (p. 332)
- [CT.CLOUDFRONT.PR.10] Require any Amazon CloudFront distributions with Amazon S3 backed origins to have origin access control configured (p. 338)
- [CT.CLOUDFRONT.PR.11] Require an Amazon CloudFront distribution to use updated SSL protocols between edge locations and custom origins (p. 345)
[CT.CLOUDFRONT.PR.1] Require an Amazon CloudFront distribution to have a default root object configured

This control checks whether an Amazon CloudFront distribution is configured to return a specific object that is the default root object.

- **Control objective**: Protect configurations
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::CloudFront::Distribution
- **AWS CloudFormation guard rule**: CT.CLOUDFRONT.PR.1 rule specification (p. 282)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CLOUDFRONT.PR.1 rule specification (p. 282)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CLOUDFRONT.PR.1 example templates (p. 284)]

Explanation

A user could possibly request a distribution's root URL instead of an object in the distribution. In this situation, specifying a default root object can help you to avoid exposing the contents of your web distribution.

Remediation for rule failure

Specify a default root object in the DefaultRootObject property.

The examples that follow show how to implement this remediation.

**Amazon CloudFront Distribution - Example**

Amazon CloudFront distribution configured with a default root object. The example is shown in JSON and in YAML.

**JSON example**

```
{
  "CloudFrontDistribution": {
    "Type": "AWS::CloudFront::Distribution",
    "Properties": {
      "DistributionConfig": {
        "Enabled": false,
        "Origins": [
          {
            "Id": "sampleOrigin",
            "DomainName": "example.com",
            "CustomOriginConfig": {
              "OriginProtocolPolicy": "https-only"
            }
          }
        ],
        "DefaultCacheBehavior": {
          "ViewerProtocolPolicy": "https-only",
```
YAML example

CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      Origins:
        - Id: sampleOrigin
          DomainName: example.com
          CustomOriginConfig:
            OriginProtocolPolicy: https-only
      DefaultCacheBehavior:
        ViewerProtocolPolicy: https-only
        TargetOriginId: sampleOrigin
        CachePolicyId: !Ref 'CachePolicy'
        DefaultRootObject: index.html

CT.CLOUDFRONT.PR.1 rule specification

# ****************************
##       Rule Specification       ##
# ****************************
#
# Rule Identifier:
#  cloudfront_default_root_object_configured_check
#
# Description:
#  This control checks whether an Amazon CloudFront distribution is configured to return a
#  specific object that is the default root object.
#
# Reports on:
#  AWS::CloudFront::Distribution
#
# Evaluates:
#  AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#  None
#
# Scenarios:
#  Scenario: 1
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#    And: The input document does not contain any CloudFront distribution resources
#    Then: SKIP
#  Scenario: 2

282
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CloudFront distribution resource
And: 'DefaultRootObject' is not present on the CloudFront distribution resource or is present and is an empty string
Then: FAIL

Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CloudFront distribution resource
And: 'DefaultRootObject' is present on the CloudFront distribution resource and is a non-empty string
Then: PASS

# Constants
let CLOUDFRONT_DISTRIBUTION_TYPE = "AWS::CloudFront::Distribution"
let INPUT_DOCUMENT = this

# Assignments
let cloudfront_distributions = Resources.*[ Type == %CLOUDFRONT_DISTRIBUTION_TYPE ]

# Primary Rules
rule cloudfront_default_root_object_configured_check when is_cfn_template(%INPUT_DOCUMENT) %cloudfront_distributions not empty {
  check(%cloudfront_distributions.Properties)
  <<
  [CT.CLOUDFRONT.PR.1]: Require an Amazon CloudFront distribution to have a default root object configured
  [FIX]: Specify a default root object in the 'DefaultRootObject' property.
  >>
}

rule cloudfront_default_root_object_configured_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDFRONT_DISTRIBUTION_TYPE) {
  check(%INPUT_DOCUMENT.%CLOUDFRONT_DISTRIBUTION_TYPE.resourceProperties)
  <<
  [CT.CLOUDFRONT.PR.1]: Require an Amazon CloudFront distribution to have a default root object configured
  [FIX]: Specify a default root object in the 'DefaultRootObject' property.
  >>
}

# Parameterized Rules
rule check(cloudfront_distribution) {
  %cloudfront_distribution {
    DistributionConfig exists
    DistributionConfig is_struct
    DistributionConfig {
      # Scenario 2
      DefaultRootObject exists
      # Scenario 3
      check_is_string_and_not_empty(DefaultRootObject)
    }
  }
}
# Utility Rules

rule check_is_string_and_not_empty(value) {
    %value {
        this is_string
        this != /\A\s*\z/
    }
}

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or 
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.CLOUDFRONT.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
- CachePolicy:
    Type: AWS::CloudFront::CachePolicy
    Properties:
        CachePolicyConfig:
            DefaultTTL: 20
            MaxTTL: 20
            MinTTL: 19
            Name:
                Fn::Sub: ${AWS::StackName}-example-cache-policy
        ParametersInCacheKeyAndForwardedToOrigin:
            CookiesConfig:
                CookieBehavior: none
                EnableAcceptEncodingGzip: false
            HeadersConfig:
                HeaderBehavior: none
            QueryStringsConfig:
                QueryStringBehavior: none
    CloudFrontDistribution:
        Type: AWS::CloudFront::Distribution
        Properties:
            DistributionConfig:
                Enabled: false
                Origins:
                    - Id: exampleOrigin
                        DomainName: example.com
                        CustomOriginConfig:
                            OriginProtocolPolicy: https-only
                            DefaultCacheBehavior:
                                ViewerProtocolPolicy: https-only
                                TargetOriginId: exampleOrigin
                                CachePolicyId:
                                    Ref: CachePolicy
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:

CachePolicy:
  Type: AWS::CloudFront::CachePolicy
  Properties:
    CachePolicyConfig:
      DefaultTTL: 20
      MaxTTL: 20
      MinTTL: 19
      Name:
        Fn::Sub: ${AWS::StackName}-example-cache-policy
  ParametersInCacheKeyAndForwardedToOrigin:
    CookiesConfig:
      CookieBehavior: none
      EnableAcceptEncodingGzip: false
      HeadersConfig:
        HeaderBehavior: none
      QueryStringsConfig:
        QueryStringBehavior: none

CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      Origins:
        - Id: exampleOrigin
          DomainName: example.com
          CustomOriginConfig:
            OriginProtocolPolicy: https-only
            DefaultCacheBehavior:
              ViewerProtocolPolicy: https-only
              TargetOriginId: exampleOrigin
          CachePolicyId:
            Ref: CachePolicy

[CT.CLOUDFRONT.PR.2] Require any Amazon CloudFront distributions with Amazon S3 backed origins to have an origin access identity configured

This control checks whether Amazon CloudFront distributions backed by Amazon S3 are configured with an origin access identity.

- Control objective: Protect configurations
- Implementation: AWS CloudFormation Guard Rule
- Control behavior: Proactive
- Resource types: AWS::CloudFront::Distribution
- AWS CloudFormation guard rule: CT.CLOUDFRONT.PR.2 rule specification (p. 287)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: CT.CLOUDFRONT.PR.2 rule specification (p. 287)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.CLOUDFRONT.PR.2 example templates (p. 291)

Explanation

CloudFront OAI prevents users from gaining direct access to Amazon S3 bucket content. With direct access to an Amazon S3 bucket, a user bypasses the CloudFront distribution and any permissions that are applied to the underlying S3 bucket content.

Usage considerations

• This control applies only to Amazon CloudFront distributions that are configured with one or more origins that are backed by Amazon S3.

Remediation for rule failure

Configure Amazon S3 backed origins by means of the Origins property. For each origin backed by Amazon S3, configure an origin access identity by means of the OriginAccessIdentity property within an S3OriginConfig configuration.

The examples that follow show how to implement this remediation.

Amazon CloudFront Distribution - Example

Amazon CloudFront distribution with an Amazon S3 bucket origin and origin access identity. The example is shown in JSON and in YAML.

JSON example

```json
{
   "CloudFrontDistribution": {
      "Type": "AWS::CloudFront::Distribution",
      "Properties": {
         "DistributionConfig": {
            "Enabled": false,
            "Origins": [
               {
                  "Id": "sampleS3Origin",
                  "DomainName": {
                     "Fn::GetAtt": [
                        "OriginBucket",
                        "RegionalDomainName"
                     ]
                  },
                  "S3OriginConfig": {
                     "OriginAccessIdentity": {
                        "Fn::Join": [
                           "",
                           ["origin-access-identity/cloudfront/",
                           {
                              "Ref": "OriginBucketOai"
                           }]
                        ]
                     }
                  }
               }
            ]
         }
      }
   }
}
```
"DefaultCacheBehavior": {
  "ViewerProtocolPolicy": "https-only",
  "TargetOriginId": "sampleS3Origin",
  "CachePolicyId": {
    "Ref": "CachePolicy"
  }
}
}

YAML example

CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      Origins:
        - Id: sampleS3Origin
          DomainName: !GetAtt 'OriginBucket.RegionalDomainName'
          S3OriginConfig:
            OriginAccessIdentity: !Join
              - '
              - - origin-access-identity/cloudfront/
              - !Ref 'OriginBucketOai'
          DefaultCacheBehavior:
            ViewerProtocolPolicy: https-only
            TargetOriginId: sampleS3Origin
            CachePolicyId: !Ref 'CachePolicy'

CT.CLOUDFRONT.PR.2 rule specification

# ##########################################################################
##       Rule Specification        ##
##########################################################################
#
# Rule Identifier:
#  cloudfront_origin_access_identity_enabled_check
#
# Description:
#  This control checks whether Amazon CloudFront distributions backed by Amazon S3 are
#  configured with an origin access identity.
#
# Reports on:
#  AWS::CloudFront::Distribution
#
# Evaluates:
#  AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#  None
#
# Scenarios:
#  Scenario: 1
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document does not contain any CloudFront distribution resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a CloudFront distribution resource
# And: No S3 backed 'Origins' are provided on the CloudFront distribution resource or 'Origins' is not present on the CloudFront distribution resource or is present and an empty list
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a CloudFront distribution resource
# And: 'S3Origin' is present on the CloudFront distribution resource
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a CloudFront distribution resource
# And: One or more S3 backed 'Origins' are configured on the CloudFront distribution resource
# And: 'OriginAccessIdentity' is not present or is an empty string in the 'S3OriginConfig' property or invalid local reference
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a CloudFront distribution resource
# And: One or more S3 backed 'Origins' are provided on the CloudFront distribution resource
# And: 'S3OriginConfig' is present with an 'OriginAccessIdentity for each S3 backed 'Origin' on the CloudFront distribution resource that is a non-empty string or valid local reference
# Then: PASS

# Constants
let CLOUDFRONT_DISTRIBUTION_TYPE = "AWS::CloudFront::Distribution"
let S3_BUCKET_DNS_NAME_PATTERN = /(.*)\.s3(-external-\d|[\-.][a-z]*-[a-z]*-[0-9])?\.amazonaws\.com(\.cn)?$/
let INPUT_DOCUMENT = this
#
# Assignments
#
let cloudfront_distributions = Resources.*[ Type == %CLOUDFRONT_DISTRIBUTION_TYPE ]
#
# Primary Rules
#
rule cloudfront_origin_access_identity_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%cloudfront_distributions not empty {
  check(%cloudfront_distributions.Properties)
  [CT.CLOUDFRONT.PR.2]: Require any Amazon CloudFront distributions with Amazon S3 backed origins to have an origin access identity configured
  [FIX]: Configure Amazon S3 backed origins by means of the 'Origins' property. For each origin backed by Amazon S3, configure an origin access identity by means of the 'OriginAccessIdentity' property within an 'S3OriginConfig' configuration.
}

288
rule cloudfront_origin_access_identity_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDFRONT_DISTRIBUTION_TYPE) {
    check(%INPUT_DOCUMENT.%CLOUDFRONT_DISTRIBUTION_TYPE.resourceProperties)
    <<
    [CT.CLOUDFRONT.PR.2]: Require any Amazon CloudFront distributions with Amazon S3 backed origins to have an origin access identity configured
    [FIX]: Configure Amazon S3 backed origins by means of the 'Origins' property. For each origin backed by Amazon S3, configure an origin access identity by means of the 'OriginAccessIdentity' property within an 'S3OriginConfig' configuration.
    >>
}

# Parameterized Rules
#
rule check(cloudfront_distribution) {
    %cloudfront_distribution[filter_cloudfront_distribution_with_legacy_s3_origins(this)] {
        DistributionConfig {
            # Scenario 3
            S3Origin not exists
        }
    }
    %cloudfront_distribution[filter_cloudfront_distribution_with_origins(this)] {
        DistributionConfig {
            # Scenario 4
            Origins [
                DomainName == %S3_BUCKET_DNS_NAME_PATTERN or
                check_origin_domain_name_get_att(DomainName)
            ] {
                S3OriginConfig exists
                S3OriginConfig is_struct
                S3OriginConfig { # Scenario 3 and 5
                    OriginAccessIdentity exists
                    check_is_string_and_not_empty(OriginAccessIdentity) or
                    check_local_oai(OriginAccessIdentity)
                }
            }
        }
    }
}

rule check_origin_domain_name_get_att(domain) {
    %domain {
        'Fn::GetAtt' {
            this is_list
            this not empty
            this[1] == "DomainName" or
            this[1] == "RegionalDomainName"
        }
        check_local_references(%INPUT_DOCUMENT, this, "AWS::S3::Bucket")
    }
}

rule check_local_oai(oai) {
    %oai {
        'Fn::Join' {
            this[1] exists
            this[1] is_list
            this[1] not empty
        }
    }
}
some this[1].* {  
  check_local_references(%INPUT_DOCUMENT, this,  
  "AWS::CloudFront::CloudFrontOriginAccessIdentity")}; 
}
} or  
'Fn::Sub' {  
  when this is_list {  
    this[1] exists  
    this[1] is_struct  
    some this[1].* {  
      check_local_references(%INPUT_DOCUMENT, this,  
      "AWS::CloudFront::CloudFrontOriginAccessIdentity")}; 
    }
  when this is_string {  
    check_is_string_and_not_empty(this) 
  }
}
}

rule filter_cloudfront_distribution_with_origins(cloudfront_distribution) {  
  %cloudfront_distribution {  
    DistributionConfig exists  
    DistributionConfig is_struct  
      DistributionConfig {  
        Origins exists  
        Origins is_list  
        Origins not empty  
      }
  }
}

rule filter_cloudfront_distribution_with_legacy_s3_origins(cloudfront_distribution) {  
  %cloudfront_distribution {  
    DistributionConfig exists  
    DistributionConfig is_struct  
      DistributionConfig {  
        S3Origin exists  
      }
  }
}

# Utility Rules  
#  
# rule check_is_string_and_not_empty(value) {  
#   %value {  
#     this is_string  
#     this != /\A\s*\z/  
#   }
# }

rule is_cfn_template(doc) {  
  %doc {  
    AWSTemplateFormatVersion exists or  
    Resources exists  
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {  
  %doc.%RESOURCE_TYPE.resourceProperties exists  
}
rule check_local_references(doc, reference_properties, referenced_resource_type) {
    %reference_properties {
        'Fn::GetAtt' {
            query_for_resource(doc, this[0], referenced_resource_type)
            <<Local Stack reference was invalid>>
        } or Ref {
            query_for_resource(doc, this, referenced_resource_type)
            <<Local Stack reference was invalid>>
        }
    }
}

rule query_for_resource(doc, resource_key, referenced_resource_type) {
    let referenced_resource = %doc.Resources[keys == resource_key]
    %referenced_resource not empty
    %referenced_resource {
        Type == referenced_resource_type
    }
}

CT.CLOUDFRONT.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
CachePolicy:
  Type: AWS::CloudFront::CachePolicy
  Properties:
  CachePolicyConfig:
    DefaultTTL: 20
    MaxTTL: 20
    MinTTL: 19
  Name:
    Fn::Sub: ${AWS::StackName}-example-cache-policy
ParametersInCacheKeyAndForwardedToOrigin:
  CookiesConfig:
    CookieBehavior: none
    EnableAcceptEncodingGzip: false
  HeadersConfig:
    HeaderBehavior: none
  QueryStringsConfig:
    QueryStringBehavior: none
OriginBucketOai:
  Type: AWS::CloudFront::CloudFrontOriginAccessIdentity
  Properties:
    CloudFrontOriginAccessIdentityConfig:
      Comment:
        Fn::Sub: ${AWS::StackName}-example-oai
OriginBucket:
  Type: AWS::S3::Bucket
OriginBucketPolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    Bucket:
      Ref: OriginBucket
    PolicyDocument:
      Version: 2012-10-17
      Statement:
        - Action:
- 's3:GetObject'
  Effect: Allow
  Resource:
  Fn::Join:
  - ''
  - 'arn:aws:s3:::'
  - Ref: OriginBucket
  - /*
  Principal:
  AWS:
  Fn::Join:
  - ''
  - 'arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity '
  - Ref: OriginBucketOai

CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
  DistributionConfig:
    Enabled: false
    Origins:
      - Id: exampleS3Origin
        DomainName:
          Fn::GetAtt:
          - OriginBucket
          - RegionalDomainName
        S3OriginConfig:
          OriginAccessIdentity:
            Fn::Join:
            - ''
            - "origin-access-identity/cloudfront/
            - Ref: OriginBucketOai
        DefaultCacheBehavior:
          ViewerProtocolPolicy: https-only
          TargetOriginId: exampleS3Origin
          CachePolicyId:
            Ref: CachePolicy

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  CachePolicy:
    Type: AWS::CloudFront::CachePolicy
    Properties:
      CachePolicyConfig:
        DefaultTTL: 20
        MaxTTL: 20
        MinTTL: 19
        Name:
          Fn::Sub: ${AWS::StackName}-example-cache-policy
    ParametersInCacheKeyAndForwardedToOrigin:
      CookiesConfig:
        CookieBehavior: none
        EnableAcceptEncodingGzip: false
      HeadersConfig:
        HeaderBehavior: none
      QueryStringsConfig:
        QueryStringBehavior: none
  CloudFrontDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Enabled: false
Proactive controls

Origins:
- Id: exampleS3Origin
  DomainName: examplebucket.s3.amazonaws.com
  S3OriginConfig: {}
DefaultCacheBehavior:
  ViewerProtocolPolicy: https-only
  TargetOriginId: exampleS3Origin
  CachePolicyId:
    Ref: CachePolicy

[CT.CLOUDFRONT.PR.3] Require an Amazon CloudFront distribution to have encryption in transit configured

This control checks whether your Amazon CloudFront distributions use HTTPS, either directly or through a redirection.

- **Control objective:** Encrypt data in transit
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CloudFront::Distribution
- **AWS CloudFormation guard rule:** [CT.CLOUDFRONT.PR.3 rule specification](p. 295)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CLOUDFRONT.PR.3 rule specification](p. 295)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CLOUDFRONT.PR.3 example templates](p. 298)

Explanation

HTTPS (TLS) can help prevent potential attackers from attempting person-in-the-middle or similar attacks, which can eavesdrop on or manipulate network traffic. Only encrypted connections over HTTPS (TLS) should be allowed. Encrypting data in transit can affect performance. We recommend that you test your application with this feature to understand the performance profile and the impact of TLS.

Remediation for rule failure

Set ViewerProtocolPolicy in DefaultCacheBehavior and CacheBehavior to https-only or redirect-to-https.

The examples that follow show how to implement this remediation.

Amazon CloudFront Distribution - Example One

Amazon CloudFront distribution configured with a default cache behavior that requires viewer connections to use HTTPS. The example is shown in JSON and in YAML.

JSON example

```json
{
    "CloudFrontDistribution": {
```
"Type": "AWS::CloudFront::Distribution",
"Properties": {
    "DistributionConfig": {
        "Enabled": false,
        "Origins": [
            {
                "Id": "sampleOrigin",
                "DomainName": "example.com",
                "CustomOriginConfig": {
                    "OriginProtocolPolicy": "https-only"
                }
            }
        ],
        "DefaultCacheBehavior": {
            "ViewerProtocolPolicy": "https-only",
            "TargetOriginId": "sampleOrigin",
            "CachePolicyId": {
                "Ref": "CachePolicy"
            }
        }
    }
}

The examples that follow show how to implement this remediation.

**Amazon CloudFront Distribution - Example Two**

Amazon CloudFront distribution configured with a cache behavior that redirects viewer HTTP connections to HTTPS. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "CloudFrontDistribution": {
        "Type": "AWS::CloudFront::Distribution",
        "Properties": {
            "DistributionConfig": {
                "Enabled": false,
```
"Origins": [  
  {  
    "Id": "sampleOrigin",  
    "DomainName": "example.com",  
    "CustomOriginConfig": {  
      "OriginProtocolPolicy": "https-only"  
    }  
  }],  
"DefaultCacheBehavior": {  
  "ViewerProtocolPolicy": "https-only",  
  "TargetOriginId": "sampleOrigin",  
  "CachePolicyId": {  
    "Ref": "CachePolicy"  
  }  
},  
"CacheBehaviors": [  
  {  
    "ViewerProtocolPolicy": "redirect-to-https",  
    "TargetOriginId": "sampleOrigin",  
    "PathPattern": "*"  
  }  
]  
}  
}  

YAML example

CloudFrontDistribution:  
  Type: AWS::CloudFront::Distribution  
  Properties:  
    DistributionConfig:  
      Enabled: false  
      Origins:  
        - Id: sampleOrigin  
          DomainName: example.com  
          CustomOriginConfig:  
            OriginProtocolPolicy: https-only  
      DefaultCacheBehavior:  
        ViewerProtocolPolicy: https-only  
        TargetOriginId: sampleOrigin  
        CachePolicyId: !Ref 'CachePolicy'  
      CacheBehaviors:  
        - ViewerProtocolPolicy: redirect-to-https  
          TargetOriginId: sampleOrigin  
          PathPattern: "*"

CT.CLOUDFRONT.PR.3 rule specification

# ****************************  
##       Rule Specification       ##  
*****************************

# Rule Identifier:  
# cloudfront_viewer_policy_https_check  
#
# Description:
# This control checks whether your Amazon CloudFront distributions use HTTPS, either
directly or through a redirection.
#
# Reports on:
# AWS::CloudFront::Distribution
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document does not contain any CloudFront distribution resources
#   Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a CloudFront distribution resource
#   And: 'DistributionConfig.DefaultCacheBehavior' is missing on the CloudFront
distribution resource
#   Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a CloudFront distribution resource
#   And: 'DistributionConfig_DEFAULTCacheBehavior' is present on the CloudFront
distribution resource
#   And: 'ViewerProtocolPolicy' in 'DefaultCacheBehavior' is missing or set to a value
#     other than 'https-only' or 'redirect-to-https' (e.g. 'allow-all')
#   Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a CloudFront distribution resource
#   And: 'DistributionConfig.CacheBehavior' is provided on the CloudFront distribution
#     resource
#   And: 'ViewerProtocolPolicy' in the 'CacheBehavior' is missing or set to a value
#     other than 'https-only' or 'redirect-to-https' (e.g. 'allow-all')
#   Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a CloudFront distribution resource
#   And: 'DistributionConfig.DefaultCacheBehavior' is present on the CloudFront
distribution resource
#   And: 'ViewerProtocolPolicy' in 'DefaultCacheBehavior' is set to 'https-only' or
#     'redirect-to-https'
#   Then: PASS
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a CloudFront distribution resource
#   And: 'DistributionConfig.CacheBehavior' are provided on the CloudFront distribution
#     resource as a non-empty list
#   And: 'ViewerProtocolPolicy' in the 'CacheBehavior' is set to 'https-only' or
#     'redirect-to-https'
#   Then: PASS
# Constants
#
let CLOUDFRONT_DISTRIBUTION_TYPE = "AWS::CloudFront::Distribution"
let ALLOWED_VIEWER_PROTOCOL_POLICIES = [ "https-only", "redirect-to-https" ]
let INPUT_DOCUMENT = this
#
# Assignments
#
let cloudfront_distributions = Resources.*[ Type == %CLOUDFRONT_DISTRIBUTION_TYPE ]
#
# Primary Rules
#
rule cloudfront_viewer_policy_https_check when is_cfn_template(%INPUT_DOCUMENT)
%cloudfront_distributions not empty {
  check(%cloudfront_distributions.Properties)
  <<
  [CT.CLOUDFRONT.PR.3]: Require an Amazon CloudFront distribution to have encryption in transit configured
  [FIX]: Set 'ViewerProtocolPolicy' in 'DefaultCacheBehavior' and 'CacheBehavior' to 'https-only' or 'redirect-to-https'.
  >>
}

rule cloudfront_viewer_policy_https_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDFRONT_DISTRIBUTION_TYPE) {
  check(%INPUT_DOCUMENT.%CLOUDFRONT_DISTRIBUTION_TYPE.resourceProperties)
  <<
  [CT.CLOUDFRONT.PR.3]: Require an Amazon CloudFront distribution to have encryption in transit configured
  [FIX]: Set 'ViewerProtocolPolicy' in 'DefaultCacheBehavior' and 'CacheBehavior' to 'https-only' or 'redirect-to-https'.
  >>
}
#
# Parameterized Rules
#
rule check(cloudfront_distribution) {
  %cloudfront_distribution {
    DistributionConfig exists
    DistributionConfig is_struct
    DistributionConfig {
      DefaultCacheBehavior exists
      DefaultCacheBehavior is_struct
      DefaultCacheBehavior {
        # Scenarios 2 and 4
        check_viewer_protocol_policy(this)
      }
    }
    when CacheBehaviors exists
    CacheBehaviors is_list
    CacheBehaviors not empty {
      CacheBehaviors[*] {
        # Scenarios 3 and 5
        check_viewer_protocol_policy(this)
      }
    }
  }
}
CT.CLOUDFRONT.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  CachePolicy:
    Type: AWS::CloudFront::CachePolicy
    Properties:
      CachePolicyConfig:
        DefaultTTL: 20
        MaxTTL: 20
        MinTTL: 19
        Name:
          Fn::Sub: ${AWS::StackName}-example-cache-policy
      ParametersInCacheKeyAndForwardedToOrigin:
        CookiesConfig:
          CookieBehavior: none
        EnableAcceptEncodingGzip: false
        HeadersConfig:
          HeaderBehavior: none
        QueryStringsConfig:
          QueryStringBehavior: none
  CloudFrontDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Enabled: false
        Origins:
          - Id: exampleOrigin
            DomainName: example.com
            CustomOriginConfig:
              OriginProtocolPolicy: https-only
        DefaultCacheBehavior:
          ViewerProtocolPolicy: https-only
      TargetOriginId: exampleOrigin
      CachePolicyId:
        Ref: CachePolicy
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
CachePolicy:
  Type: AWS::CloudFront::CachePolicy
  Properties:
    CachePolicyConfig:
      DefaultTTL: 20
      MaxTTL: 20
      MinTTL: 19
      Name:
        Fn::Sub: ${AWS::StackName}-example-cache-policy
  ParametersInCacheKeyAndForwardedToOrigin:
    CookiesConfig:
      CookieBehavior: none
    EnableAcceptEncodingGzip: false
    HeadersConfig:
      HeaderBehavior: none
    QueryStringsConfig:
      QueryStringBehavior: none
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      Origins:
        - Id: exampleOrigin
          DomainName: example.com
          CustomOriginConfig:
            OriginProtocolPolicy: https-only
            DefaultCacheBehavior:
              ViewerProtocolPolicy: allow-all
              TargetOriginId: exampleOrigin
              CachePolicyId:
                Ref: CachePolicy

[CT.CLOUDFRONT.PR.4] Require an Amazon CloudFront distribution to have origin failover configured

This control checks whether your Amazon CloudFront distribution is configured with an origin group that contains two origin group members.

- **Control objective**: Improve availability
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::CloudFront::Distribution
- **AWS CloudFormation guard rule**: [CT.CLOUDFRONT.PR.4 rule specification](p. 301)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CLOUDFRONT.PR.4 rule specification](p. 301)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CLOUDFRONT.PR.4 example templates](p. 304)
Explanation

CloudFront origin failover can increase availability. Origin failover automatically redirects traffic to a secondary origin if the primary origin is unavailable or if it returns specific HTTP response status codes.

Remediation for rule failure

Configure an origin group on the Amazon CloudFront Distribution with two origin group members.

The examples that follow show how to implement this remediation.

Amazon CloudFront Distribution - Example One

Amazon CloudFront distribution configured with an origin group that contains two origin group members. The example is shown in JSON and in YAML.

JSON example

```json
{
    "CloudFrontDistribution": {
        "Type": "AWS::CloudFront::Distribution",
        "Properties": {
            "DistributionConfig": {
                "Enabled": false,
                "Origins": [
                    {
                        "Id": "sampleOrigin",
                        "DomainName": "one.example.com",
                        "CustomOriginConfig": {
                            "OriginProtocolPolicy": "https-only"
                        }
                    },
                    {
                        "Id": "sampleOrigin2",
                        "DomainName": "two.example.com",
                        "CustomOriginConfig": {
                            "OriginProtocolPolicy": "https-only"
                        }
                    }
                ],
                "DefaultCacheBehavior": {
                    "ViewerProtocolPolicy": "https-only",
                    "TargetOriginId": "sampleOrigin",
                    "CachePolicyId": {
                        "Ref": "CachePolicy"
                    }
                },
                "OriginGroups": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Id": "ExampleOriginGroup",
                            "FailoverCriteria": {
                                "StatusCodes": {
                                    "Items": [
                                        400
                                    ],
                                    "Quantity": 1
                                }
                            },
                            "Members": {
                                "Quantity": 2,
                                "Items": [
```
YAML example

```yaml
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      Origins:
        - Id: sampleOrigin
          DomainName: one.example.com
          CustomOriginConfig:
            OriginProtocolPolicy: https-only
        - Id: sampleOrigin2
          DomainName: two.example.com
          CustomOriginConfig:
            OriginProtocolPolicy: https-only
      DefaultCacheBehavior:
        ViewerProtocolPolicy: https-only
        TargetOriginId: sampleOrigin
        CachePolicyId: !Ref 'CachePolicy'
      OriginGroups:
        Quantity: 1
        Items:
          - Id: ExampleOriginGroup
            FailoverCriteria:
              StatusCodes:
                Items:
                - 400
                Quantity: 1
              Members:
                Quantity: 2
                Items:
                - OriginId: sampleOrigin
                - OriginId: sampleOrigin2
```

CT.CLOUDFRONT.PR.4 rule specification

```plaintext
# ###################################################################
##       Rule Specification       ##
###################################################################
#
# Rule Identifier:
```
## cloudfront_origin_failover_enabled_check

### Description:
This control checks whether your Amazon CloudFront distribution is configured with an origin group that contains two origin group members.

### Reports on:
AWS::CloudFront::Distribution

### Evaluates:
AWS CloudFormation, AWS CloudFormation hook

### Rule Parameters:
None

### Scenarios:

#### Scenario: 1
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   And: The input document does not contain any CloudFront distribution resources
   Then: SKIP

#### Scenario: 2
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   And: The input document contains a CloudFront distribution resource
   And: 'OriginGroups' is not present on the CloudFront distribution resource
   Then: FAIL

#### Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   And: The input document contains a CloudFront distribution resource
   And: 'OriginGroups' is present on the CloudFront distribution resource
   And: 'Quantity' within 'OriginGroups' is 0
   Then: FAIL

#### Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   And: The input document contains a CloudFront distribution resource
   And: 'OriginGroups' is present on the CloudFront distribution resource
   And: 'Quantity' within 'OriginGroups' is >= 1
   And: 'Quantity' within 'Members' is < 2
   Then: FAIL

#### Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   And: The input document contains a CloudFront distribution resource
   And: 'OriginGroups' is present on the CloudFront distribution resource
   And: 'Quantity' within 'OriginGroups' is >= 1
   And: 'Quantity' within 'Members' is == 2
   Then: PASS

### Constants

let CLOUDFRONT_DISTRIBUTION_TYPE = "AWS::CloudFront::Distribution"
let INPUT_DOCUMENT = this

### Assignments

let cloudfront_distributions = Resources.*[ Type == %CLOUDFRONT_DISTRIBUTION_TYPE ]

### Primary Rules

rule cloudfront_origin_failover_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%cloudfront_distributions not empty {
  check(%cloudfront_distributions.Properties) <<
  [CT.CLOUDFRONT.PR.4]: Require an Amazon CloudFront distribution to have origin failover configured
  [FIX]: Configure an origin group on the Amazon CloudFront Distribution with two origin group members.
  >>
}

rule cloudfront_origin_failover_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDFRONT_DISTRIBUTION_TYPE) {
  check(%INPUT_DOCUMENT.%CLOUDFRONT_DISTRIBUTION_TYPE.resourceProperties) <<
  [CT.CLOUDFRONT.PR.4]: Require an Amazon CloudFront distribution to have origin failover configured
  [FIX]: Configure an origin group on the Amazon CloudFront Distribution with two origin group members.
  >>
}

# Parameterized Rules
#
rule check(cloudfront_distribution) {
  %cloudfront_distribution {
    DistributionConfig exists
    DistributionConfig is_struct
    DistributionConfig {
      # Scenario 2
      OriginGroups exists
      OriginGroups is_struct
      OriginGroups {
        # Scenario 3
        Quantity exists
        Quantity >= 1
        Items exists
        Items is_list
        Items not empty
        Items[*] {
          Members exists
          Members is_struct
          Members {
            # Scenarios 4 and 5
            Quantity == 2
          }
        }
      }
    }
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or Resources exists
  }
}
CT.CLOUDFRONT.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
- CachePolicy:
  - Type: AWS::CloudFront::CachePolicy
  - Properties:
    - CachePolicyConfig:
      - DefaultTTL: 20
      - MaxTTL: 20
      - MinTTL: 19
      - Name: Fn::Sub: ${AWS::StackName}-example-cache-policy
      - ParametersInCacheKeyAndForwardedToOrigin:
        - CookiesConfig:
          - CookieBehavior: none
          - EnableAcceptEncodingGzip: false
        - HeadersConfig:
          - HeaderBehavior: none
        - QueryStringsConfig:
          - QueryStringBehavior: none
  - CloudFrontDistribution:
    - Type: AWS::CloudFront::Distribution
    - Properties:
      - DistributionConfig:
        - Enabled: false
        - Origins:
        - Id: exampleOrigin
          - DomainName: one.example.com
          - CustomOriginConfig:
            - OriginProtocolPolicy: https-only
        - Id: exampleOrigin2
          - DomainName: two.example.com
          - CustomOriginConfig:
            - OriginProtocolPolicy: https-only
      - DefaultCacheBehavior:
        - ViewerProtocolPolicy: https-only
        - TargetOriginId: exampleOrigin
      - CachePolicyId:
        - Ref: CachePolicy
  - OriginGroups:
    - Quantity: 1
    - Items:
      - Id: ExampleOriginGroup
      - FailoverCriteria:
        - StatusCodes:
          - Quantity: 1
          - Items:
            - 400
      - Members:
        - Quantity: 2
        - Items:
          - OriginId: exampleOrigin
          - OriginId: exampleOrigin2
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
    CachePolicy:
        Type: AWS::CloudFront::CachePolicy
        Properties:
            CachePolicyConfig:
                DefaultTTL: 20
                MaxTTL: 20
                MinTTL: 19
                Name:
                    Fn::Sub: ${AWS::StackName}-example-cache-policy
            ParametersInCacheKeyAndForwardedToOrigin:
                CookiesConfig:
                    CookieBehavior: none
                    EnableAcceptEncodingGzip: false
                    HeadersConfig:
                        HeaderBehavior: none
                    QueryStringsConfig:
                        QueryStringBehavior: none
    CloudFrontDistribution:
        Type: AWS::CloudFront::Distribution
        Properties:
            DistributionConfig:
                Enabled: false
                Origins:
                    - Id: exampleOrigin
                        DomainName: example.com
                        CustomOriginConfig:
                            OriginProtocolPolicy: https-only
                            DefaultCacheBehavior:
                                ViewerProtocolPolicy: https-only
                                TargetOriginId: exampleOrigin
                                CachePolicyId:
                                    Ref: CachePolicy

[CT.CLOUDFRONT.PR.5] Require any Amazon CloudFront distribution to have logging enabled

This control checks whether Amazon CloudFront distributions are configured with access logging.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CloudFront::Distribution
- **AWS CloudFormation guard rule:** [CT.CLOUDFRONT.PR.5 rule specification](p. 307)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CLOUDFRONT.PR.5 rule specification](p. 307)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CLOUDFRONT.PR.5 example templates](p. 309)
**Explanation**

CloudFront access logs provide detailed information about every user request that CloudFront receives. Each log contains information such as the date and time the request was received, the IP address of the viewer that made the request, the source of the request, and the port number of the request from the viewer.

These access logs are useful for applications such as security and access audits, and in forensic investigation.

**Remediation for rule failure**

Set `Bucket` in `DistributionConfig.Logging` to an Amazon S3 bucket that has been configured to receive Amazon CloudFront distribution access logs.

The examples that follow show how to implement this remediation.

**Amazon CloudFront Distribution - Example**

Amazon CloudFront distribution configured with access logging enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "CloudFrontDistribution": {
    "Type": "AWS::CloudFront::Distribution",
    "Properties": {
      "DistributionConfig": {
        "Enabled": false,
        "Origins": [
          {
            "Id": "sampleOrigin",
            "DomainName": "example.com",
            "CustomOriginConfig": {
              "OriginProtocolPolicy": "https-only"
            }
          }
        ],
        "DefaultCacheBehavior": {
          "ViewerProtocolPolicy": "https-only",
          "TargetOriginId": "sampleOrigin",
          "CachePolicyId": {
            "Ref": "CachePolicy"
          }
        },
        "Logging": {
          "Bucket": {
            "Fn::GetAtt": [
              "LoggingBucket",
              "RegionalDomainName"
            ]
          }
        }
      }
    }
  }
}
```

**YAML example**

```yaml
306
```
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      Origins:
        - Id: sampleOrigin
          DomainName: example.com
          CustomOriginConfig:
            OriginProtocolPolicy: https-only
        DefaultCacheBehavior:
          ViewerProtocolPolicy: https-only
          TargetOriginId: sampleOrigin
          CachePolicyId: !Ref 'CachePolicy'
      Logging:
        Bucket: !GetAtt 'LoggingBucket.RegionalDomainName'

CT.CLOUDFRONT.PR.5 rule specification

# ####################################################################
## Rule Specification       ##
# #####################################################################
#
# Rule Identifier:
#   cloudfront_access_logs_enabled_check
#
# Description:
#   This control checks whether Amazon CloudFront distributions are configured with access
#   logging.
#
# Reports on:
#   AWS::CloudFront::Distribution
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any CloudFront distribution resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document contains a CloudFront distribution resource
#     And: 'DistributionConfig.Logging.Bucket' configuration is not present on the
#     CloudFront distribution resource
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a CloudFront distribution resource
#     And: 'DistributionConfig.Logging' configuration is present on the CloudFront
distribution resource
#     And: 'Bucket' has been provided in the 'DistributionConfig.Logging' configuration
#     with with an empty string or
# Scenario 4

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>The input document is an AWS CloudFormation or AWS CloudFormation hook document and contains a CloudFront distribution resource.</td>
</tr>
<tr>
<td></td>
<td>A 'DistributionConfig.Logging' configuration is present on the CloudFront distribution resource.</td>
</tr>
<tr>
<td></td>
<td>A 'Bucket' property has been provided within the 'DistributionConfig.Logging' configuration with a non-empty string or valid local stack reference.</td>
</tr>
<tr>
<td></td>
<td>Then: PASS</td>
</tr>
</tbody>
</table>

Constants

let CLOUDFRONT_DISTRIBUTION_TYPE = "AWS::CloudFront::Distribution"
let S3_BUCKET_TYPE = "AWS::S3::Bucket"
let INPUT_DOCUMENT = this

Assignments

let cloudfront_distributions = Resources.*[ Type == %CLOUDFRONT_DISTRIBUTION_TYPE ]

Primary Rules

rule cloudfront_access_logs_enabled_check when is_cfn_template(%INPUT_DOCUMENT) {
    %cloudfront_distributions not empty {
        check(%cloudfront_distributions.Properties)
        <<
        [CT.CLOUDFRONT.PR.5]: Require any Amazon CloudFront distribution to have logging enabled
        [FIX]: Set 'Bucket' in 'DistributionConfig.Logging' to an Amazon S3 bucket that has been configured to receive Amazon CloudFront distribution access logs.
        >>
    }
}

rule cloudfront_access_logs_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDFRONT_DISTRIBUTION_TYPE) {
    check(%INPUT_DOCUMENT.%CLOUDFRONT_DISTRIBUTION_TYPE.resourceProperties)
    <<
    [CT.CLOUDFRONT.PR.5]: Require any Amazon CloudFront distribution to have logging enabled
    [FIX]: Set 'Bucket' in 'DistributionConfig.Logging' to an Amazon S3 bucket that has been configured to receive Amazon CloudFront distribution access logs.
    >>
}

Parameterized Rules

rule check(cloudfront_distribution) {
    %cloudfront_distribution {
        DistributionConfig exists
        DistributionConfig is_struct
        DistributionConfig {
            Logging exists
            Logging is_struct
            Logging {
                # Scenario 2
                Bucket exists
                # Scenarios 3 and 4
            }
        }
    }
}
CT.CLOUDFRONT.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  CachePolicy:
    Type: AWS::CloudFront::CachePolicy
    Properties:
      CachePolicyConfig:
        DefaultTTL: 20
        MaxTTL: 20
        MinTTL: 19
```
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
CachePolicy:
  Type: AWS::CloudFront::CachePolicy
  Properties:
    CachePolicyConfig:
      DefaultTTL: 20
Proactive controls

MaxTTL: 20
MinTTL: 19
Name:
  Fn::Sub: ${AWS::StackName}-example-cache-policy

ParametersInCacheKeyAndForwardedToOrigin:
  CookiesConfig:
    CookieBehavior: none
  EnableAcceptEncodingGzip: false
  HeadersConfig:
    HeaderBehavior: none
  QueryStringsConfig:
    QueryStringBehavior: none

CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
    Origins:
      - Id: exampleOrigin
        DomainName: example.com
        CustomOriginConfig:
          OriginProtocolPolicy: https-only
        DefaultCacheBehavior:
          ViewerProtocolPolicy: https-only
        TargetOriginId: exampleOrigin
    CachePolicyId:
      Ref: CachePolicy

[CT.CLOUDFRONT.PR.6] Require an Amazon CloudFront distribution to use custom SSL/TLS certificates

This control checks whether the certificate associated with an Amazon CloudFront distribution is a custom SSL/TLS certificate.

- **Control objective:** Encrypt data in transit
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CloudFront::Distribution
- **AWS CloudFormation guard rule:** [CT.CLOUDFRONT.PR.6 rule specification](p. 313)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CLOUDFRONT.PR.6 rule specification](p. 313)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CLOUDFRONT.PR.6 example templates](p. 316)

Explanation

Custom SSL/TLS certificates give your users access to content by using alternate domain names. You can store custom certificates in AWS Certificate Manager (recommended), or in IAM.

Usage considerations

- This control requires a viewer certificate configuration compatible only with Amazon CloudFront distributions that use Aliases, also known as alternate domain names or CNAMEs.
Remediation for rule failure


The examples that follow show how to implement this remediation.

Amazon CloudFront Distribution - Example

Amazon CloudFront distribution configured with an AWS Certificate Manager SSL certificate. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "CloudFrontDistribution": {
    "Type": "AWS::CloudFront::Distribution",
    "Properties": {
      "DistributionConfig": {
        "Enabled": false,
        "Origins": [
          {
            "Id": "sampleOrigin",
            "DomainName": "example.com",
            "CustomOriginConfig": {
              "OriginProtocolPolicy": "https-only"
            }
          }
        ],
        "DefaultCacheBehavior": {
          "ViewerProtocolPolicy": "https-only",
          "TargetOriginId": "sampleOrigin",
          "CachePolicyId": {
            "Ref": "CachePolicy"
          }
        },
        "ViewerCertificate": {
          "AcmCertificateArn": {
            "Ref": "ACMCertificate"
          },
          "MinimumProtocolVersion": "TLSv1.2_2021",
          "SslSupportMethod": "sni-only"
        }
      }
    }
  }
}
```

**YAML example**

```yaml
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      Origins:
      - Id: sampleOrigin
        DomainName: example.com
        CustomOriginConfig:
          OriginProtocolPolicy: https-only
```

312
DefaultCacheBehavior:
  ViewerProtocolPolicy: https-only
  TargetOriginId: sampleOrigin
  CachePolicyId: !Ref 'CachePolicy'
ViewerCertificate:
  AcmCertificateArn: !Ref 'ACMCertificate'
  MinimumProtocolVersion: TLSv1.2_2021
  SslSupportMethod: sni-only

CT.CLOUDFRONT.PR.6 rule specification

# #################################################################################
#  Rule Specification          #
#  #################################################################################
#
# Rule Identifier:
#  cloudfront_custom_ssl_certificate_check
#
# Description:
#  This control checks whether the certificate associated with an Amazon CloudFront
#  distribution is a custom SSL/TLS certificate.
#
# Reports on:
#  AWS::CloudFront::Distribution
#
# Evaluates:
#  AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#  None
#
# Scenarios:
#  Scenario: 1
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#    And: The input document does not contain any CloudFront distribution resources
#    Then: SKIP
#  Scenario: 2
#    Given: The input document contains a CloudFront distribution resource
#    And: 'ViewerCertificate' is not present on the CloudFront distribution resource
#    Then: FAIL
#  Scenario: 3
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#    And: The input document contains a CloudFront distribution resource
#    And: 'ViewerCertificate' is present on the CloudFront distribution resource
#    And: 'CloudFrontDefaultCertificate' is set to bool(true)
#    Then: FAIL
#  Scenario: 4
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#    And: The input document contains a CloudFront distribution resource
#    And: 'ViewerCertificate' is present on the CloudFront distribution resource
#    And: One of 'AcmCertificateArn' or 'IamCertificateId' are not provided or provided
#      as empty strings or invalid local references
#    And: One of 'MinimumProtocolVersion' and 'SslSupportMethod' is not provided or
#      provided as an empty string
#    Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a CloudFront distribution resource
# And: 'ViewerCertificate' is present on the CloudFront distribution resource
# And: 'AcmCertificateArn' or 'IamCertificateId' are provided in the
# 'ViewerCertificate' configuration as
#   non-empty strings or 'AcmCertificateArn' is a valid local reference
# And: 'MinimumProtocolVersion' and 'SslSupportMethod' are provided as non-empty
#   strings
# Then: PASS
#
# Constants
#
let CLOUDFRONT_DISTRIBUTION_TYPE = "AWS::CloudFront::Distribution"
let INPUT_DOCUMENT = this
#
# Assignments
#
let cloudfront_distributions = Resources.*[ Type == %CLOUDFRONT_DISTRIBUTION_TYPE ]
#
# Primary Rules
#
rule cloudfront_custom_ssl_certificate_check when is_cfn_template(%INPUT_DOCUMENT)
  %cloudfront_distributions not empty {
    check(%cloudfront_distributions.Properties)
    <<
    [CT.CLOUDFRONT.PR.6]: Require an Amazon CloudFront distribution to use custom SSL/
    TLS certificates
    [FIX]: Provide a 'ViewerCertificate' configuration with values for
    'AcmCertificateArn', 'MinimumProtocolVersion', and 'SslSupportMethod'.
    >>
  }
rule cloudfront_custom_ssl_certificate_check when is_cfn_hook(%INPUT_DOCUMENT,
  %CLOUDFRONT_DISTRIBUTION_TYPE) {
  check(%INPUT_DOCUMENT.%CLOUDFRONT_DISTRIBUTION_TYPE.resourceProperties)
  <<
  [CT.CLOUDFRONT.PR.6]: Require an Amazon CloudFront distribution to use custom SSL/
  TLS certificates
  [FIX]: Provide a 'ViewerCertificate' configuration with values for
  'AcmCertificateArn', 'MinimumProtocolVersion', and 'SslSupportMethod'.
  >>
}
#
# Parameterized Rules
#
rule check(cloudfront_distribution) {
  %cloudfront_distribution {
    DistributionConfig exists
    DistributionConfig is_struct
    
    DistributionConfig {
      ViewerCertificate exists
      ViewerCertificate is_struct
      
      ViewerCertificate {
        CloudFrontDefaultCertificate not exists or
        CloudFrontDefaultCertificate == false
        check_custom_acm_certificate_provided(AcmCertificateArn,
          "AWS::CertificateManager::Certificate") or
check_custom_iam_certificate_provided(IamCertificateId)
MinimumProtocolVersion exists
check_is_string_and_not_empty(MinimumProtocolVersion)
SslSupportMethod exists
check_is_string_and_not_empty(SslSupportMethod)

rule check_custom_acm_certificate_provided(certificate, cfn_type) {
  %certificate {
    this exists
    check_is_string_and_not_empty(this) or
    check_local_references(%INPUT_DOCUMENT, this, %cfn_type)
  }
}

rule check_custom_iam_certificate_provided(certificate) {
  %certificate {
    this exists
    check_is_string_and_not_empty(this)
  }
}

# Utility Rules
#
rule check_is_string_and_not_empty(value) {
  %value {
    this is_string
    this != /\A\s*\z/}
}

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

rule check_local_references(doc, reference_properties, referenced_resource_type) {
  %reference_properties {
    'Fn::GetAtt' {
      query_for_resource(%doc, this[0], %referenced_resource_type)
      <<Local Stack reference was invalid>>
    } or Ref {
      query_for_resource(%doc, this, %referenced_resource_type)
      <<Local Stack reference was invalid>>
    }
  }
}

rule query_for_resource(doc, resource_key, referenced_resource_type) {
  let referenced_resource = %doc.Resources[ keys == %resource_key ]
  %referenced_resource not empty
  %referenced_resource {
    Type == %referenced_resource_type
  }
}
CT.CLOUDFRONT.PR.6 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```json
Resources:
  CachePolicy:
    Type: AWS::CloudFront::CachePolicy
    Properties:
      CachePolicyConfig:
        DefaultTTL: 20
        MaxTTL: 20
        MinTTL: 19
        Name:
          Fn::Sub: ${AWS::StackName}-example-cache-policy
      ParametersInCacheKeyAndForwardedToOrigin:
        CookiesConfig:
          CookieBehavior: none
          EnableAcceptEncodingGzip: false
        HeadersConfig:
          HeaderBehavior: none
        QueryStringsConfig:
          QueryStringBehavior: none
    ACMCertificate:
      Type: "AWS::CertificateManager::Certificate"
      Properties:
        DomainName: example.com
        ValidationMethod: DNS
        DomainValidationOptions:
          - DomainName: www.example.com
          HostedZoneId: ZZZHHHHWWWWAAA
    CloudFrontDistribution:
      Type: AWS::CloudFront::Distribution
      Properties:
        DistributionConfig:
          Enabled: false
        Origins:
          - Id: exampleOrigin
            DomainName: example.com
            CustomOriginConfig:
              OriginProtocolPolicy: https-only
        DefaultCacheBehavior:
          ViewerProtocolPolicy: https-only
          TargetOriginId: exampleOrigin
          CachePolicyId:
            Ref: CachePolicy
        ViewerCertificate:
          AcmCertificateArn:
            Ref: ACMCertificate
          MinimumProtocolVersion: TLSv1.2_2021
          SslSupportMethod: sni-only
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.
[CT.CLOUDFRONT.PR.7] Require an Amazon CloudFront distribution to use SNI to serve HTTPS requests

This control checks whether your Amazon CloudFront distributions are configured to use SNI to serve HTTPS requests.

- **Control objective**: Encrypt data in transit, Improve availability
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::CloudFront::Distribution
- **AWS CloudFormation guard rule**: CT.CLOUDFRONT.PR.7 rule specification (p. 319)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.CLOUDFRONT.PR.7 rule specification (p. 319)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.CLOUDFRONT.PR.7 example templates (p. 322)

Explanation
Server Name Indication (SNI) is an extension to the TLS protocol. It is supported by browsers and clients released after 2010. If you configure CloudFront to serve HTTPS requests using SNI, CloudFront associates your alternate domain name with an IP address for each edge location. When a viewer submits an HTTPS request for your content, DNS routes the request to the IP address for the correct edge location. The IP address for your domain name is determined during the SSL/TLS handshake negotiation; the IP address isn't dedicated to your distribution.

**Usage considerations**

- This control requires a viewer certificate configuration which is only compatible with Amazon CloudFront distributions that use Aliases (also known as alternate domain names or CNAMEs)

**Remediation for rule failure**

Within ViewerCertificate, set SslSupportMethod to sni-only, MinimumProtocolVersion to a protocol that supports SNI (TLSv1 or greater), and AcmCertificateArn to the ARN of an AWS ACM certificate.

The examples that follow show how to implement this remediation.

**Amazon CloudFront Distribution - Example**

Amazon CloudFront distribution configured to use SNI to serve HTTPS requests. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "CloudFrontDistribution": {
      "Type": "AWS::CloudFront::Distribution",
      "Properties": {
         "DistributionConfig": {
            "Enabled": false,
            "Origins": [
               {
                  "Id": "sampleOrigin",
                  "DomainName": "example.com",
                  "CustomOriginConfig": {
                     "OriginProtocolPolicy": "https-only"
                  }
               }
            ],
            "DefaultCacheBehavior": {
               "ViewerProtocolPolicy": "https-only",
               "TargetOriginId": "sampleOrigin",
               "CachePolicyId": {
                  "Ref": "CachePolicy"
               }
            },
            "ViewerCertificate": {
               "AcmCertificateArn": {
                  "Ref": "ACMCertificate"
               },
               "MinimumProtocolVersion": "TLSv1.2_2021",
               "SslSupportMethod": "sni-only"
            }
         }
      }
   }
}
```
YAML example

```yaml
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      Origins:
        - Id: sampleOrigin
          DomainName: example.com
          CustomOriginConfig:
            OriginProtocolPolicy: https-only
    DefaultCacheBehavior:
      ViewerProtocolPolicy: https-only
      TargetOriginId: sampleOrigin
      CachePolicyId: !Ref 'CachePolicy'
    ViewerCertificate:
      AcmCertificateArn: !Ref 'ACMCertificate'
      MinimumProtocolVersion: TLSv1.2_2021
      SslSupportMethod: sni-only
```

CT.CLOUDFRONT.PR.7 rule specification

```plaintext
# ##############################################################################
#                          Rule Specification                          ############
# ##############################################################################
# # Rule Identifier:
# # cloudfront_sni_enabled_check
# # Description:
# # This control checks whether your Amazon CloudFront distributions are configured to use
# # SNI to serve HTTPS requests.
# # Reports on:
# # AWS::CloudFront::Distribution
# # Evaluates:
# # AWS CloudFormation, AWS CloudFormation hook
# # Rule Parameters:
# # None
# # Scenarios:
# # Scenario: 1
# # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
# # document
# # And: The input document does not contain any CloudFront distribution resources
# # Then: SKIP
# # Scenario: 2
# # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
# # document
# # And: The input document contains a CloudFront distribution resource
# # And: 'ViewerCertificate' is not present on the CloudFront distribution resource
# # Then: FAIL
# # Scenario: 3
```
# Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CloudFront distribution resource
And: 'ViewerCertificate' is present on the CloudFront distribution resource
And: 'CloudFrontDefaultCertificate' is set to bool(true)
Then: FAIL

# Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CloudFront distribution resource
And: 'ViewerCertificate' is present on the CloudFront distribution resource
And: 'AcmCertificateArn' or 'IamCertificateId' are provided in the 'ViewerCertificate' configuration
And: 'MinimumProtocolVersion' is provided in the 'ViewerCertificate' configuration
with a protocol that does not support SNI (SSLv3)
Then: FAIL

# Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CloudFront distribution resource
And: 'ViewerCertificate' is present on the CloudFront distribution resource
And: 'AcmCertificateArn' or 'IamCertificateId' are provided in the 'ViewerCertificate' configuration
And: 'MinimumProtocolVersion' is provided in the 'ViewerCertificate' configuration
with a protocol that supports SNI (TLSv1 or greater)
And: 'SslSupportMethod' is set to 'vip'
Then: FAIL

# Scenario: 7
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CloudFront distribution resource
And: 'ViewerCertificate' is present on the CloudFront distribution resource
And: 'AcmCertificateArn' or 'IamCertificateId' are provided in the 'ViewerCertificate' configuration
And: 'MinimumProtocolVersion' is provided in the 'ViewerCertificate' configuration
with a protocol that supports SNI (TLSv1 or greater)
And: 'SslSupportMethod' is set to 'sni-only'
Then: PASS

# Constants
let CLOUDFRONT_DISTRIBUTION_TYPE = "AWS::CloudFront::Distribution"
let UNSUPPORTED_PROTOCOLS_FOR_SNI = ['SSLv3']
let INPUT_DOCUMENT = this

# Assignments
let cloudfront_distributions = Resources.*[ Type == %CLOUDFRONT_DISTRIBUTION_TYPE ]

# Primary Rules
rule cloudfront_sni_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%cloudfront_distributions not empty {
  check(%cloudfront_distributions.Properties)
  <=
  [CT.CLOUDFRONT.PR.7]: Require an Amazon CloudFront distribution to use SNI to serve HTTPS requests
  [FIX]: Within 'ViewerCertificate', set 'SslSupportMethod' to 'sni-only', 'MinimumProtocolVersion' to a protocol that supports SNI ('TLSv1' or greater), and 'AcmCertificateArn' to the ARN of an AWS ACM certificate.
rule cloudfront_sni_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDFRONT_DISTRIBUTION_TYPE) {
    check(%INPUT_DOCUMENT.%CLOUDFRONT_DISTRIBUTION_TYPE.resourceProperties) <<
    [CT.CLOUDFRONT.PR.7]: Require an Amazon CloudFront distribution to use SNI to serve HTTPS requests
    [FIX]: Within 'ViewerCertificate', set 'SslSupportMethod' to 'sni-only', 'MinimumProtocolVersion' to a protocol that supports SNI ('TLSv1' or greater), and 'AcmCertificateArn' to the ARN of an AWS ACM certificate.
} >>

# Parameterized Rules
#
rule check(cloudfront_distribution) {
    %cloudfront_distribution {
        DistributionConfig exists
        DistributionConfig is_struct
        DistributionConfig {
            ViewerCertificate exists
            ViewerCertificate is_struct
            ViewerCertificate {
                CloudFrontDefaultCertificate not exists or
                CloudFrontDefaultCertificate == false
                check_custom_acm_certificate_provided(AcmCertificateArn, "AWS::CertificateManager::Certificate") or
                check_custom_iam_certificate_provided(IamCertificateId)
                MinimumProtocolVersion exists
                MinimumProtocolVersion not in %UNSUPPORTED_PROTOCOLS_FOR_SNI
                SslSupportMethod exists
                SslSupportMethod == "sni-only"
            }
        }
    }
} >>

rule check_custom_acm_certificate_provided(certificate, cfn_type) {
    %certificate {
        this exists
        check_is_string_and_not_empty(this) or
        check_local_references(%INPUT_DOCUMENT, this, %cfn_type)
    }
} >>

rule check_custom_iam_certificate_provided(certificate) {
    %certificate {
        this exists
        check_is_string_and_not_empty(this)
    }
} >>

# Utility Rules
#
rule check_is_string_and_not_empty(value) {
    %value {
        this is_string
    }
} >>
CT.CLOUDFRONT.PR.7 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
- CachePolicy:
  - Type: AWS::CloudFront::CachePolicy
  - Properties:
    - CachePolicyConfig:
      - DefaultTTL: 20
      - MaxTTL: 20
      - MinTTL: 19
      - Name:
        - Fn::Sub: ${AWS::StackName}-example-cache-policy
      - ParametersInCacheKeyAndForwadedToOrigin:
        - CookiesConfig:
          - CookieBehavior: none
          - EnableAcceptEncodingGzip: false
        - HeadersConfig:
          - HeaderBehavior: none
        - QueryStringsConfig:
          - QueryStringBehavior: none
      - ACMCertificate:
        - Type: "AWS::CertificateManager::Certificate"
Proactive controls

Properties:
  DomainName: example.com
  ValidationMethod: DNS
  DomainValidationOptions:
    - DomainName: www.example.com
    - HostedZoneId: ZZZHHHHWWWWAAA

CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      Origins:
        - Id: exampleOrigin
          DomainName: example.com
          CustomOriginConfig:
            OriginProtocolPolicy: https-only
            DefaultCacheBehavior:
              ViewerProtocolPolicy: https-only
              TargetOriginId: exampleOrigin
              CachePolicyId:
                Ref: CachePolicy
              ViewerCertificate:
                AcmCertificateArn:
                  Ref: ACMCertificate
                MinimumProtocolVersion: TLSv1.2_2021
                SslSupportMethod: sni-only

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  CachePolicy:
    Type: AWS::CloudFront::CachePolicy
    Properties:
      CachePolicyConfig:
        DefaultTTL: 20
        MaxTTL: 20
        MinTTL: 19
        Name:
          Fn::Sub: ${AWS::StackName}-example-cache-policy
      ParametersInCacheKeyAndForwardedToOrigin:
        CookiesConfig:
          CookieBehavior: none
          EnableAcceptEncodingGzip: false
        HeadersConfig:
          HeaderBehavior: none
        QueryStringsConfig:
          QueryStringBehavior: none
  ACMCertificate:
    Type: "AWS::CertificateManager::Certificate"
    Properties:
      DomainName: example.com
      ValidationMethod: DNS
      DomainValidationOptions:
        - DomainName: www.example.com
        - HostedZoneId: ZZZHHHHWWWWAAA
  CloudFrontDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Enabled: false
        Origins:
          - Id: exampleOrigin
CT.CLOUDFRONT.PR.8] Require an Amazon CloudFront distribution to encrypt traffic to custom origins

This control checks whether your Amazon CloudFront distributions are encrypting traffic to custom origins.

- **Control objective:** Encrypt data in transit
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CloudFront::Distribution
- **AWS CloudFormation guard rule:** [CT.CLOUDFRONT.PR.8 rule specification](p. 327)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CLOUDFRONT.PR.8 rule specification](p. 327)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CLOUDFRONT.PR.8 example templates](p. 330)

Explanation

HTTPS (TLS) can help prevent eavesdropping or manipulation of network traffic. Only encrypted connections over HTTPS (TLS) should be allowed.

**Usage considerations**

- This control applies only to Amazon CloudFront distributions that have one or more origins configured.

Remediation for rule failure

For Amazon CloudFront custom origins, set OriginProtocolPolicy to `https-only` or `match-viewer`. When setting OriginProtocolPolicy to `match-viewer`, do not set ViewerProtocolPolicy to `allow-all` for any cache behaviors.

The examples that follow show how to implement this remediation.

**Amazon CloudFront Distribution - Example One**

Amazon CloudFront distribution configured to require HTTPS connections to custom origins, by means of an origin protocol policy of `https-only`. The example is shown in JSON and in YAML.
JSON example

```json
{
  "CloudFrontDistribution": {
    "Type": "AWS::CloudFront::Distribution",
    "Properties": {
      "DistributionConfig": {
        "Enabled": false,
        "DefaultCacheBehavior": {
          "ViewerProtocolPolicy": "https-only",
          "TargetOriginId": "sampleOrigin",
          "CachePolicyId": {
            "Ref": "CachePolicy"
          }
        }
      },
      "CacheBehaviors": [
        {
          "ViewerProtocolPolicy": "https-only",
          "TargetOriginId": "sampleOrigin",
          "PathPattern": "*",
          "CachePolicyId": {
            "Ref": "CachePolicy"
          }
        }
      ],
      "Origins": [
        {
          "Id": "sampleOrigin",
          "DomainName": "example.com",
          "CustomOriginConfig": {
            "OriginProtocolPolicy": "https-only"
          }
        }
      ]
    }
  }
}
```

YAML example

```yaml
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      DefaultCacheBehavior:
        ViewerProtocolPolicy: https-only
        TargetOriginId: sampleOrigin
        CachePolicyId: !Ref 'CachePolicy'
    CacheBehaviors:
      - ViewerProtocolPolicy: https-only
        TargetOriginId: sampleOrigin
        PathPattern: "*"
        CachePolicyId: !Ref 'CachePolicy'
    Origins:
      - Id: sampleOrigin
        DomainName: example.com
        CustomOriginConfig:
          OriginProtocolPolicy: https-only
```
The examples that follow show how to implement this remediation.

**Amazon CloudFront Distribution - Example Two**

Amazon CloudFront distribution configured to require HTTPS connections to custom origins, by means of an origin protocol policy of match-viewer. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "CloudFrontDistribution": {
        "Type": "AWS::CloudFront::Distribution",
        "Properties": {
            "DistributionConfig": {
                "Enabled": false,
                "DefaultCacheBehavior": {
                    "ViewerProtocolPolicy": "https-only",
                    "TargetOriginId": "sampleOrigin",
                    "CachePolicyId": {
                        "Ref": "CachePolicy"
                    }
                },
                "CacheBehaviors": [
                    {
                        "ViewerProtocolPolicy": "https-only",
                        "TargetOriginId": "sampleOrigin",
                        "PathPattern": "*",
                        "CachePolicyId": {
                            "Ref": "CachePolicy"
                        }
                    }
                ],
                "Origins": [
                    {
                        "Id": "sampleOrigin",
                        "DomainName": "example.com",
                        "CustomOriginConfig": {
                            "OriginProtocolPolicy": "match-viewer"
                        }
                    }
                ]
            }
        }
    }
}
```

**YAML example**

```yaml
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      DefaultCacheBehavior:
        ViewerProtocolPolicy: https-only
        TargetOriginId: sampleOrigin
        CachePolicyId: !Ref 'CachePolicy'
    CacheBehaviors:
```

326
- ViewerProtocolPolicy: https-only
  TargetOriginId: sampleOrigin
  PathPattern: ‘*’
  CachePolicyId: !Ref 'CachePolicy'

Origins:
- Id: sampleOrigin
  DomainName: example.com
  CustomOriginConfig:
    OriginProtocolPolicy: match-viewer

---

CT.CLOUDFRONT.PR.8 rule specification

```
# ###################################################################
##       Rule Specification        ##
# ###################################################################
#
# Rule Identifier:
#   cloudfront_traffic_to_origin_encrypted_check
#
# Description:
#   This control checks whether your Amazon CloudFront distributions are encrypting traffic
to custom origins.
#
# Reports on:
#   AWS::CloudFront::Distribution
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any CloudFront distribution resources
#     Then: SKIPs
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a CloudFront distribution resource
#     And: 'Origins' is not present or is an empty list
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a CloudFront distribution resource
#     And: One or more 'Origins' has been configured
#     And: There are no 'Origins' with a 'CustomOriginConfig'
#     Then: SKIP
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a CloudFront distribution resource
#     And: 'CustomOrigin' is present on the CloudFront distribution resource
#     Then: FAIL
#   Scenario: 5
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a CloudFront distribution resource
```
# And: One or more 'Origins' has been configured
# And: There one or more 'Origins' with a 'CustomOriginConfig'
# And: At least one 'Origins' with a 'CustomOriginConfig' has an 'OriginProtocolPolicy' of 'http-only'
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a CloudFront distribution resource
# And: One or more 'Origins' has been configured
# And: There one or more 'Origins' with a 'CustomOriginConfig'
# And: At least one 'Origins' with a 'CustomOriginConfig' has an 'OriginProtocolPolicy' of 'match-viewer'
# And: Any 'ViewerProtocolPolicy' is set to 'allow-all' for 'DefaultCacheBehavior' or any configured
# 'CacheBehaviors'
# Then: FAIL
# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a CloudFront distribution resource
# And: One or more 'Origins' has been configured
# And: There one or more 'Origins' with a 'CustomOriginConfig'
# And: All 'Origins' with a 'CustomOriginConfig' have an 'OriginProtocolPolicy' of 'https-only'
# Then: PASS
# Scenario: 8
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a CloudFront Distribution resource
# And: One or more 'Origins' has been configured
# And: There one or more 'Origins' with a 'CustomOriginConfig'
# And: At least one 'Origins' with a 'CustomOriginConfig' has an 'OriginProtocolPolicy' of 'match-viewer'
# And: 'ViewerProtocolPolicy' is not set to 'allow-all' for both 'DefaultCacheBehavior' and any configured
# 'CacheBehaviors'
# Then: PASS

# Constants
#
let CLOUDFRONT_DISTRIBUTION_TYPE = "AWS::CloudFront::Distribution"
let INPUT_DOCUMENT = this

# Assignments
#
let cloudfront_distributions = Resources.*[ Type == %CLOUDFRONT_DISTRIBUTION_TYPE ]

# Primary Rules
#
rule cloudfront_traffic_to_origin_encrypted_check when is_cfn_template(%INPUT_DOCUMENT)
%cloudfront_distributions not empty
{
    check(%cloudfront_distributions.Properties)
    <<
        [CT.CLOUDFRONT.PR.8]: Require an Amazon CloudFront distribution to encrypt traffic to custom origins
        [FIX]: For Amazon CloudFront custom origins, set 'OriginProtocolPolicy' to 'https-only' or match-viewer'. When setting 'OriginProtocolPolicy' to 'match-viewer', do not set 'ViewerProtocolPolicy' to 'allow-all' for any cache behaviors.
    >>
}
rule cloudfront_traffic_to_origin_encrypted_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDFRONT DISTRIBUTION TYPE) {
  check(%INPUT DOCUMENT, %CLOUDFRONT DISTRIBUTION TYPE.resourceProperties)
<<
  [CT.CLOUDFRONT.PR.8]: Require an Amazon CloudFront distribution to encrypt traffic to custom origins
  [FIX]: For Amazon CloudFront custom origins, set 'OriginProtocolPolicy' to 'https-only' or 'match-viewer'. When setting 'OriginProtocolPolicy' to 'match-viewer', do not set 'ViewerProtocolPolicy' to 'allow-all' for any cache behaviors.
>>
}

# Parameterized Rules
#
# rule check(cloudfront_distribution) {
%cloudfront_distribution[
  filter_cloudfront_distribution_with_legacy_origins(this)
] {
  DistributionConfig {
    # Scenario 4
    CustomOrigin not exists
  }
}

%cloudfront_distribution[
  # Scenario 2
  filter_cloudfront_distribution_with_origins(this)
] {
  let cloudfront_distro = this
  DistributionConfig {
    Origins [
      # Scenario 3
      CustomOriginConfig exists
      CustomOriginConfig is_struct
    ] {
      CustomOriginConfig {
        # Scenario 5
        OriginProtocolPolicy != "http-only"
        # Scenario 6
        OriginProtocolPolicy == "https-only" or
        # Scenario 6 and 8
        match_viewer_policy_with_no_allow_all_viewer_protocol_policy(OriginProtocolPolicy,
        %cloudfront_distro)
      }
    }
  }
}
}

rule match_viewer_policy_with_no_allow_all_viewer_protocol_policy(origin_protocol_policy, cloudfront_distribution) {
  %origin_protocol_policy {
    this == "match-viewer"
    %cloudfront_distribution {
      DistributionConfig {
        DefaultCacheBehavior exists
        DefaultCacheBehavior is_struct
        DefaultCacheBehavior {
          check_viewer_protocol_policy(this)
        }
      }
    }
  }
}
CT.CLOUDFRONT.PR.8 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.
Resources:
  CachePolicy:
    Type: AWS::CloudFront::CachePolicy
    Properties:
      CachePolicyConfig:
        DefaultTTL: 20
        MaxTTL: 20
        MinTTL: 19
        Name:
          Fn::Sub: ${AWS::StackName}-example-cache-policy
      ParametersInCacheKeyAndForwardedToOrigin:
        CookiesConfig:
          CookieBehavior: none
          EnableAcceptEncodingGzip: false
        HeadersConfig:
          HeaderBehavior: none
        QueryStringsConfig:
          QueryStringBehavior: none
  CloudFrontDistribution:
    Type: AWS::CloudFront::Distribution
    Properties:
      DistributionConfig:
        Enabled: false
        DefaultCacheBehavior:
          ViewerProtocolPolicy: https-only
          TargetOriginId: exampleOrigin
          CachePolicyId:
            Ref: CachePolicy
        CacheBehaviors:
          - ViewerProtocolPolicy: https-only
            TargetOriginId: exampleOrigin
            PathPattern: '*'
            CachePolicyId:
              Ref: CachePolicy
        Origins:
          - Id: exampleOrigin
            DomainName: example.com
            CustomOriginConfig:
              OriginProtocolPolicy: https-only

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  CachePolicy:
    Type: AWS::CloudFront::CachePolicy
    Properties:
      CachePolicyConfig:
        DefaultTTL: 20
        MaxTTL: 20
        MinTTL: 19
        Name:
          Fn::Sub: ${AWS::StackName}-example-cache-policy
      ParametersInCacheKeyAndForwardedToOrigin:
        CookiesConfig:
          CookieBehavior: none
          EnableAcceptEncodingGzip: false
        HeadersConfig:
          HeaderBehavior: none
        QueryStringsConfig:
          QueryStringBehavior: none
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      DefaultCacheBehavior:
        ViewerProtocolPolicy: allow-all
        TargetOriginId: exampleOrigin
        CachePolicyId:
          Ref: CachePolicy
      Origins:
        - Id: exampleOrigin
          DomainName: example.com
          CustomOriginConfig:
            OriginProtocolPolicy: match-viewer

[CT.CLOUDFRONT.PR.9] Require an Amazon CloudFront distribution to have a security policy of TLSv1.2 as a minimum

This control checks whether your Amazon CloudFront distributions are using a minimum security policy and cipher suite of TLSv1.2 or greater for viewer connections.

- Control objective: Manage vulnerabilities
- Implementation: AWS CloudFormation Guard Rule
- Control behavior: Proactive
- Resource types: AWS::CloudFront::Distribution
- AWS CloudFormation guard rule: CT.CLOUDFRONT.PR.9 rule specification (p. 334)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.CLOUDFRONT.PR.9 rule specification (p. 334)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.CLOUDFRONT.PR.9 example templates (p. 336)

Explanation

AWS Control Tower recommends that you use SSL/TLS to communicate with AWS resources. We recommend TLS version 1.2 or later.

You can specify the security policy CloudFront will use for HTTPS connections with viewers. The security policy determines two settings: 1) the minimum SSL/TLS protocol that CloudFront can use to communicate with viewers, and 2) the ciphers that CloudFront can use to encrypt the content that it returns to viewers.

Usage considerations

- This control requires a viewer certificate configuration compatible only with Amazon CloudFront distributions that use Aliases, also known as alternate domain names or CNAMEs.

Remediation for rule failure

Provide a ViewerCertificate configuration with MinimumProtocolVersion set to TLSv1.2 or higher.
The examples that follow show how to implement this remediation.

Amazon CloudFront Distribution - Example

Amazon CloudFront distribution configured to use TLS version 1.2 for viewer HTTPS connections. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "CloudFrontDistribution": {
        "Type": "AWS::CloudFront::Distribution",
        "Properties": {
            "DistributionConfig": {
                "Enabled": false,
                "Origins": [
                    {
                        "Id": "sampleOrigin",
                        "DomainName": "example.com",
                        "CustomOriginConfig": {
                            "OriginProtocolPolicy": "https-only"
                        }
                    }
                ],
                "DefaultCacheBehavior": {
                    "ViewerProtocolPolicy": "https-only",
                    "TargetOriginId": "sampleOrigin",
                    "CachePolicyId": {
                        "Ref": "CachePolicy"
                    }
                },
                "ViewerCertificate": {
                    "MinimumProtocolVersion": "TLSv1.2_2018",
                    "AcmCertificateArn": {
                        "Ref": "ACMCertificate"
                    },
                    "SslSupportMethod": "vip"
                }
            }
        }
    }
}
```

**YAML example**

```yaml
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      Origins:
        - Id: sampleOrigin
          DomainName: example.com
          CustomOriginConfig:
            OriginProtocolPolicy: https-only
    DefaultCacheBehavior:
      ViewerProtocolPolicy: https-only
      TargetOriginId: sampleOrigin
      CachePolicyId: !Ref 'CachePolicy'
    ViewerCertificate:
```

333
Proactive controls

MinimumProtocolVersion: TLSv1.2_2018
AcmCertificateArn: !Ref 'ACMCertificate'
SslSupportMethod: vip

CT.CLOUDFRONT.PR.9 rule specification

# #####################################################################
##       Rule Specification       ##
#####################################################################
#
# Rule Identifier:
#  cloudFront_security_policy_check
#
# Description:
#  This control checks whether your Amazon CloudFront distributions are using a minimum
security policy and cipher suite of TLSv1.2 or greater for viewer connections.
#
# Reports on:
#  AWS::CloudFront::Distribution
#
# Evaluates:
#  AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#  None
#
# Scenarios:
#  Scenario: 1
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#    And: The input document does not contain any CloudFront distribution resources
#    Then: SKIP
#  Scenario: 2
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#    And: The input document contains a CloudFront distribution resource
#    And: 'DistributionConfig.ViewerCertificate' is not present on the CloudFront
distribution resource
#    Then: FAIL
#  Scenario: 3
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#    And: The input document contains a CloudFront distribution resource
#    And: 'DistributionConfig.ViewerCertificate' is present on the CloudFront
distribution resource
#    And: 'CloudFrontDefaultCertificate' in 'ViewerCertificate' is set to bool(true)
#    Then: FAIL
#  Scenario: 4
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#    And: The input document contains a CloudFront distribution resource
#    And: 'DistributionConfig.ViewerCertificate' is present on the CloudFront
distribution resource
#    And: 'CloudFrontDefaultCertificate' is not provided in 'ViewerCertificate' or
provided and set to bool(false)
#    And: 'MinimumProtocolVersion' is not provided in 'ViewerCertificate' or provided as
an empty string
#    Then: FAIL
#  Scenario: 5
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document contains a CloudFront distribution resource
And: 'DistributionConfig.ViewerCertificate' is present on the CloudFront distribution resource
And: 'CloudFrontDefaultCertificate' is not provided in 'ViewerCertificate' or provided and set to bool(false)
And: 'MinimumProtocolVersion' is provided in 'ViewerCertificate' and is to one of SSLv3, TLSv1,
      TLSv1_2016, or TLSv1.1_2016
Then: FAIL
Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CloudFront distribution resource
And: 'DistributionConfig.ViewerCertificate' is present on the CloudFront distribution resource
And: 'CloudFrontDefaultCertificate' is not provided in 'ViewerCertificate' or provided and set to bool(false)
And: 'MinimumProtocolVersion' is provided in 'ViewerCertificate' and is not set to SSLv3, TLSv1,
      TLSv1_2016, or TLSv1.1_2016
Then: PASS

Constants
let CLOUDFRONT_DISTRIBUTION_TYPE = "AWS::CloudFront::Distribution"
let INPUT_DOCUMENT = this
let NON_COMPLIANT_TLS_POLICIES_LIST = ["SSLv3", "TLSv1", "TLSv1_2016", "TLSv1.1_2016"]

Assignments
let cloudfront_distributions = Resources.[ Type == %CLOUDFRONT_DISTRIBUTION_TYPE ]

Primary Rules
# rule cloudfront_security_policy_check when is_cfn_template(%INPUT_DOCUMENT)
%cloudfront_distributions not empty {
  check(%cloudfront_distributions.Properties)
  
  [CT.CLOUDFRONT.PR.9]: Require an Amazon CloudFront distribution to have a security policy of TLSv1.2 as a minimum
  [FIX]: Provide a 'ViewerCertificate' configuration with 'MinimumProtocolVersion' set to TLSv1.2 or higher.
}

rule cloudfront_security_policy_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDFRONT_DISTRIBUTION_TYPE) {
  check(%INPUT_DOCUMENT,%CLOUDFRONT_DISTRIBUTION_TYPE.resourceProperties)
  
  [CT.CLOUDFRONT.PR.9]: Require an Amazon CloudFront distribution to have a security policy of TLSv1.2 as a minimum
  [FIX]: Provide a 'ViewerCertificate' configuration with 'MinimumProtocolVersion' set to TLSv1.2 or higher.
}

Parameterized Rules
# rule check(cloudfront_distribution) {
  %cloudfront_distribution {
    DistributionConfig exists
    DistributionConfig is_struct
  }
}
DistributionConfig {
    # Scenario 2
    ViewerCertificate exists
    ViewerCertificate is_struct

    ViewerCertificate {
        # Scenario 3
        CloudFrontDefaultCertificate not exists or
        CloudFrontDefaultCertificate == false

        # Scenario 4, 5 and 6
        MinimumProtocolVersion exists

        check_is_string_and_not_empty(MinimumProtocolVersion)
        MinimumProtocolVersion not in %NON_COMPLIANT_TLS_POLICIES_LIST
    }
}

# Utility Rules

rule check_is_string_and_not_empty(value) {
    %value {
        this is_string
        this !~ /\A\s*\z/
    }
}

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  CachePolicy:
    Type: AWS::CloudFront::CachePolicy
    Properties:
      CachePolicyConfig:
        DefaultTTL: 20
        MaxTTL: 20
        MinTTL: 19
        Name:
          Fn::Sub: ${AWS::StackName}-example-cache-policy
    ParametersInCacheKeyAndForwardedToOrigin:
      CookiesConfig:
        CookieBehavior: none
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
CachePolicy:
  Type: AWS::CloudFront::CachePolicy
  Properties:
    CachePolicyConfig:
      DefaultTTL: 20
      MaxTTL: 20
      MinTTL: 19
      Name:
        Fn::Sub: ${AWS::StackName}-example-cache-policy
    ParametersInCacheKeyAndForwardedToOrigin:
      CookiesConfig:
        CookieBehavior: none
        EnableAcceptEncodingGzip: false
        HeadersConfig:
          HeaderBehavior: none
        QueryStringsConfig:
          QueryStringBehavior: none
    ACMCertificate:
      Type: "AWS::CertificateManager::Certificate"
      Properties:
        DomainName: example.com
        ValidationMethod: DNS
        DomainValidationOptions:
          - DomainName: www.example.com
        HostedZoneId: ZZZHHHHWWWWAAA
      CloudFrontDistribution:
        Type: AWS::CloudFront::Distribution
        Properties:
          DistributionConfig:
            Enabled: false
            Origins:
              - Id: exampleOrigin
              DomainName: example.com
            CustomOriginConfig:
              OriginProtocolPolicy: https-only
            DefaultCacheBehavior:
              ViewerProtocolPolicy: https-only
              TargetOriginId: exampleOrigin
              CachePolicyId:
                Ref: CachePolicy
              ViewerCertificate:
                MinimumProtocolVersion: TLSv1.2_2018
                AcmCertificateArn:
                  Ref: ACMCertificate
                SslSupportMethod: sni-only
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      Origins:
        - Id: exampleOrigin
          DomainName: example.com
          CustomOriginConfig:
            OriginProtocolPolicy: https-only
          DefaultCacheBehavior:
            ViewerProtocolPolicy: https-only
            TargetOriginId: exampleOrigin
            CachePolicyId:
              Ref: CachePolicy
          ViewerCertificate:
            MinimumProtocolVersion: TLSv1
            AcmCertificateArn:
              Ref: ACMCertificate
            SslSupportMethod: vip

[CT.CLOUDFRONT.PR.10] Require any Amazon CloudFront distributions with Amazon S3 backed origins to have origin access control configured

This control checks whether your Amazon CloudFront distributions backed by Amazon S3 are configured to use an origin access control.

- **Control objective:** Enforce least privilege, Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CloudFront::Distribution
- **AWS CloudFormation guard rule:** CT.CLOUDFRONT.PR.10 rule specification (p. 340)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.CLOUDFRONT.PR.10 rule specification (p. 340)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.CLOUDFRONT.PR.10 example templates (p. 343)

Explanation

CloudFront OAC prevents users from gaining direct access to an Amazon S3 bucket's content. Direct access an S3 bucket bypasses the CloudFront distribution and any permissions that are applied to the underlying S3 bucket content.

Usage considerations

- This control applies only to Amazon CloudFront distributions that have one or more origins backed by Amazon S3 configured.

Remediation for rule failure

The Origins property configures origins backed by Amazon S3. For each origin backed by Amazon S3, configure an origin access control identifier using the OriginAccessControlId property.
The examples that follow show how to implement this remediation.

**Amazon CloudFront Distribution - Example**

Amazon CloudFront distribution with an Amazon S3 bucket origin, configured with an origin access control. The example is shown in JSON and in YAML.

**JSON example**

```
{
   "CloudFrontDistribution": {
      "Type": "AWS::CloudFront::Distribution",
      "Properties": {
         "DistributionConfig": {
            "Enabled": false,
            "Origins": [
               {
                  "Id": "sampleOrigin",
                  "DomainName": {
                     "Fn::GetAtt": [
                        "OriginBucket",
                        "RegionalDomainName"
                     ]
                  },
                  "OriginAccessControlId": {
                     "Ref": "OriginAccessControl"
                  },
                  "S3OriginConfig": {}
               }
            ],
            "DefaultCacheBehavior": {
               "ViewerProtocolPolicy": "https-only",
               "TargetOriginId": "sampleOrigin",
               "CachePolicyId": {
                  "Ref": "CachePolicy"
               }
            }
         }
      }
   }
}
```

**YAML example**

```
CloudFrontDistribution:
   Type: AWS::CloudFront::Distribution
   Properties:
      DistributionConfig:
         Enabled: false
         Origins:
            - Id: sampleOrigin
              DomainName: !GetAtt 'OriginBucket.RegionalDomainName'
              OriginAccessControlId: !Ref 'OriginAccessControl'
              S3OriginConfig: {}
      DefaultCacheBehavior:
         ViewerProtocolPolicy: https-only
         TargetOriginId: sampleOrigin
         CachePolicyId: !Ref 'CachePolicy'
```
CT.CLOUDFRONT.PR.10 rule specification

```
# #############################################################################
##       Rule Specification        
# #############################################################################
# Rule Identifier:
#   cloudfront_origin_access_control_enabled_check
#
# Description:
#   This control checks whether your Amazon CloudFront distributions backed by Amazon S3
#   are configured to use an origin access control.
#
# Reports on:
#   AWS::CloudFront::Distribution
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#         And: The input document does not contain any CloudFront distribution resources
#         Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#         And: The input document contains a CloudFront distribution resource
#         And: No S3 backed 'Origins' are provided on the CloudFront distribution resource or
#               'Origins' is not present on
#               the CloudFront distribution resource or is present and an empty list
#         Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#         And: The input document contains a CloudFront distribution resource
#         And: 'S3Origin' is present on the CloudFront distribution resource
#         Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#         And: The input document contains a CloudFront distribution resource
#         And: One or more S3 backed 'Origins' are configured on the CloudFront distribution
#               resource
#         And: 'OriginAccessControlId' is not present for the 'Origin' or is an empty string
#               or invalid local reference
#         Then: FAIL
#   Scenario: 5
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#         And: The input document contains a CloudFront distribution resource
#         And: One or more S3 backed 'Origins' are provided on the CloudFront distribution
#               resource
#         And: 'OriginAccessControlId' is present for each S3 backed 'Origin' and is a non-
#               empty string or valid local
#               reference
#         Then: PASS
#
# Constants
```

340
let CLOUDFRONT_DISTRIBUTION_TYPE = "AWS::CloudFront::Distribution"
let CLOUDFRONT_ORIGIN_ACCESS_CONTROL_TYPE = "AWS::CloudFront::OriginAccessControl"
let S3_BUCKET_DNS_NAME_PATTERN = /(.*)\.s3\(-external\-d|[^\-][a-z]*-[a-z]*-[0-9])?\amazonaws\-com\.(cn)?$/
let INPUT_DOCUMENT = this

let cloudfront_distributions = Resources.*[ Type == %CLOUDFRONT_DISTRIBUTION_TYPE ]

rule cloudfront_origin_access_control_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%cloudfront_distributions not empty {
  check(%cloudfront_distributions.Properties)
  %cloudfront_distributions not empty {
    [CT.CLOUDFRONT.PR.10]: Require any Amazon CloudFront distributions with Amazon S3 backed origins to have origin access control configured
    [FIX]: The 'Origins' property configures origins backed by Amazon S3. For each origin backed by Amazon S3, configure an origin access control identifier using the 'OriginAccessControlId' property.
  }
}

rule cloudfront_origin_access_control_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDFRONT_DISTRIBUTION_TYPE) {
  check(%INPUT_DOCUMENT.%CLOUDFRONT_DISTRIBUTION_TYPE.resourceProperties)
  %cloudfront_distributions not empty {
    [CT.CLOUDFRONT.PR.10]: Require any Amazon CloudFront distributions with Amazon S3 backed origins to have origin access control configured
    [FIX]: The 'Origins' property configures origins backed by Amazon S3. For each origin backed by Amazon S3, configure an origin access control identifier using the 'OriginAccessControlId' property.
  }
}

rule check(cloudfront_distribution) {
  %cloudfront_distribution[
  filter_cloudfront_distribution_with_legacy_s3_origins(this)
  ] {
    DistributionConfig {
      # Scenario 3
      S3Origin not exists
    }
  }
  %cloudfront_distribution[
  # Scenario 2
  filter_cloudfront_distribution_with_origins(this)
  ] {
    DistributionConfig {
      Origins [
        # Scenario 4
        DomainName == %S3_BUCKET_DNS_NAME_PATTERN or
        check_origin_domain_name_get_att(DomainName)
      ] {
        # Scenario 3 and 5
        OriginAccessControlId exists
        check_is_string_and_not_empty(OriginAccessControlId) or

'Fn::GetAtt' {
    query_for_resource(%doc, this[0], %referenced_resource_type)
    <<Local Stack reference was invalid>>
} or Ref {
    query_for_resource(%doc, this, %referenced_resource_type)
    <<Local Stack reference was invalid>>
}

rule query_for_resource(doc, resource_key, referenced_resource_type) {
    let referenced_resource = %doc.Resources[ keys == %resource_key ]
    %referenced_resource not empty
    %referenced_resource {
        Type == %referenced_resource_type
    }
}

CT.CLOUDFRONT.PR.10 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  CachePolicy:
    Type: AWS::CloudFront::CachePolicy
    Properties:
    CachePolicyConfig:
      DefaultTTL: 20
      MaxTTL: 20
      MinTTL: 19
      Name:
        Fn::Sub: ${AWS::StackName}-example-cache-policy
    ParametersInCacheKeyAndForwardedToOrigin:
      CookiesConfig:
        CookieBehavior: none
        EnableAcceptEncodingGzip: false
      HeadersConfig:
        HeaderBehavior: none
      QueryStringsConfig:
        QueryStringBehavior: none
    OriginAccessControl:
      Type: AWS::CloudFront::OriginAccessControl
      Properties:
      OriginAccessControlConfig:
        Name:
          Fn::Sub: ${AWS::StackName}-example-oac
        OriginAccessControlOriginType: s3
        SigningBehavior: always
        SigningProtocol: sigv4
    OriginBucket:
      Type: AWS::S3::Bucket
    OriginBucketPolicy:
      Type: AWS::S3::BucketPolicy
      Properties:
      Bucket:
        Ref: OriginBucket
      PolicyDocument:
        Version: 2012-10-17
        Statement:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  CachePolicy:
    Type: AWS::CloudFront::CachePolicy
    Properties:
      CachePolicyConfig:
        DefaultTTL: 20
        MaxTTL: 20
        MinTTL: 19
        Name:
          Fn::Sub: ${AWS::StackName}-example-cache-policy
  ParametersInCacheKeyAndForwaredToOrigin:
    CookiesConfig:
      CookieBehavior: none
      EnableAcceptEncodingGzip: false
    HeadersConfig:
      HeaderBehavior: none
    QueryStringsConfig:
      QueryStringBehavior: none
  OriginBucketOai:
    Type: AWS::CloudFront::CloudFrontOriginAccessIdentity
[CT.CLOUDFRONT.PR.11] Require an Amazon CloudFront distribution to use updated SSL protocols between edge locations and custom origins

This control checks whether your Amazon CloudFront distributions are using deprecated SSL protocols for HTTPS communication between CloudFront edge locations and custom origins.

- **Control objective:** Manage vulnerabilities
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CloudFront::Distribution
- **AWS CloudFormation guard rule:** [CT.CLOUDFRONT.PR.11 rule specification (p. 347)]
Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.CLOUDFRONT.PR.11 rule specification (p. 347)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.CLOUDFRONT.PR.11 example templates (p. 350)

Explanation

In 2015, the Internet Engineering Task Force (IETF) officially announced that SSL 3.0 should be deprecated, because the protocol is insufficiently secure. We recommend that you use TLSv1.2 or later for HTTPS communication to your custom origins.

Usage considerations

- This control applies only to Amazon CloudFront distributions that have one or more custom origins configured.

Remediation for rule failure

Remove deprecated SSL protocols from OriginSSLProtocols in Origins that have CustomOriginConfig configurations.

The examples that follow show how to implement this remediation.

Amazon CloudFront Distribution - Example

Amazon CloudFront distribution configured to use TLS v1.2 as an origin SSL protocol. The example is shown in JSON and in YAML.

JSON example

```json
{
  "CloudFrontDistribution": {
    "Type": "AWS::CloudFront::Distribution",
    "Properties": {
      "DistributionConfig": {
        "Enabled": false,
        "DefaultCacheBehavior": {
          "ViewerProtocolPolicy": "https-only",
          "TargetOriginId": "sampleOrigin",
          "CachePolicyId": {
            "Ref": "CachePolicy"
          }
        }
      },
      "Origins": [
        {
          "Id": "sampleOrigin",
          "DomainName": "example.com",
          "CustomOriginConfig": {
            "OriginProtocolPolicy": "https-only",
            "OriginSSLProtocols": [
              "TLSv1.2"
            ]
          }
        }
      ]
    }
  }
}
```
YAML example

CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      DefaultCacheBehavior:
        ViewerProtocolPolicy: https-only
        TargetOriginId: sampleOrigin
        CachePolicyId: !Ref 'CachePolicy'
    Origins:
      - Id: sampleOrigin
        DomainName: example.com
        CustomOriginConfig:
          OriginProtocolPolicy: https-only
          OriginSSLProtocols:
            - TLSv1.2

CT.CLOUDFRONT.PR.11 rule specification

# ###########################################################################
## Rule Specification    ##
###########################################################################
#
# Rule Identifier:
#   cloudfront_no_deprecated_ssl_protocols_check
#
# Description:
#   This control checks whether your Amazon CloudFront distributions are using deprecated
#   SSL protocols for HTTPS communication between CloudFront edge locations and custom
#   origins.
#
# Reports on:
#   AWS::CloudFront::Distribution
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#     And: The input document does not contain any CloudFront distribution resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#     And: The input document contains a CloudFront distribution resource
#     And: 'Origins' is not present or is an empty list
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#
# And: The input document contains a CloudFront distribution resource
# And: One or more 'Origins' has been configured
# And: There are no 'Origins' with a 'CustomOriginConfig'
# Then: SKIP
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a CloudFront distribution resource
# And: One or more 'Origins' has been configured
# And: There one or more 'Origins' with a 'CustomOriginConfig'
# And: All 'Origins' with a 'CustomOriginConfig' have an 'OriginProtocolPolicy' of 'http-only'
# Then: SKIP
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a CloudFront distribution resource
# And: 'CustomOrigin' is present on the CloudFront distribution resource
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a CloudFront distribution resource
# And: One or more 'Origins' has been configured
# And: There one or more 'Origins' with a 'CustomOriginConfig'
# And: One or more 'Origins' with a 'CustomOriginConfig' have an 'OriginProtocolPolicy' not equal to 'http-only'
# And: 'OriginSSLProtocols' has not been specified or specified as an empty list
# Then: FAIL
# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a CloudFront distribution resource
# And: One or more 'Origins' has been configured
# And: There one or more 'Origins' with a 'CustomOriginConfig'
# And: One or more 'Origins' with a 'CustomOriginConfig' have an 'OriginProtocolPolicy' not equal to 'http-only'
# And: 'OriginSSLProtocols' has been specified as a non-empty list and contains 'SSLv3'
# Then: FAIL
# Scenario: 8
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a CloudFront distribution resource
# And: One or more 'Origins' has been configured
# And: There one or more 'Origins' with a 'CustomOriginConfig'
# And: One or more 'Origins' with a 'CustomOriginConfig' have an 'OriginProtocolPolicy' not equal to 'http-only'
# And: 'OriginSSLProtocols' has been specified as a non-empty list and does not contain 'SSLv3'
# Then: PASS

# Constants
let CLOUDFRONT_DISTRIBUTION_TYPE = "AWS::CloudFront::Distribution"
let UNSUPPORTED_ORIGIN_SSL_PROTOCOLS = [ "SSLv3" ]
let OUT_OF_SCOPE_PROTOCOL_POLICIES = [ "http-only" ]
let INPUT_DOCUMENT = this

# Assignments
let cloudfront_distributions = Resources.*[ Type == %CLOUDFRONT_DISTRIBUTION_TYPE ]
# Primary Rules

```
rule cloudfront_no_deprecated_ssl_protocols_check when is_cfn_template(%INPUT_DOCUMENT)
  %cloudfront_distributions not empty
  {  
    check(%cloudfront_distributions.Properties)
    <<
    [CT.CLOUDFRONT.PR.11]: Require an Amazon CloudFront distribution to use updated SSL protocols between edge locations and custom origins
    [FIX]: Remove deprecated SSL protocols from 'OriginSSLProtocols' in 'Origins' that have 'CustomOriginConfig' configurations.
    >>
  }
```

```
rule cloudfront_no_deprecated_ssl_protocols_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDFRONT_DISTRIBUTION_TYPE) {
  check(%INPUT_DOCUMENT.%CLOUDFRONT_DISTRIBUTION_TYPE.resourceProperties)
  <<
  [CT.CLOUDFRONT.PR.11]: Require an Amazon CloudFront distribution to use updated SSL protocols between edge locations and custom origins
  [FIX]: Remove deprecated SSL protocols from 'OriginSSLProtocols' in 'Origins' that have 'CustomOriginConfig' configurations.
  >>
}
```

# Parameterized Rules

```
rule check(cloudfront_distribution) {
  %cloudfront_distribution[
    filter_cloudfront_distribution_with_legacy_origins(this)
  ] {
    DistributionConfig {
      # Scenario 5
      CustomOrigin not exists
    }
  }

  %cloudfront_distribution[
    # Scenario 2
    filter_cloudfront_distribution_with_origins(this)
  ] {
    DistributionConfig {
      Origins [
        # Scenario 3 and 4
        CustomOriginConfig exists
        CustomOriginConfig is_struct
        filter_custom_origin_config(CustomOriginConfig)
      ] {
        CustomOriginConfig {
          # Scenario 6, 7 and 8
          OriginSSLProtocols exists
          OriginSSLProtocols is_list
          OriginSSLProtocols not empty
          %UNSUPPORTED_ORIGIN_SSL_PROTOCOLS.* not in OriginSSLProtocols
        }
      }
    }
  }
}
```

```
rule filter_cloudfront_distribution_with_origins(cloudfront_distribution) {
  %cloudfront_distribution {
    DistributionConfig exists
    DistributionConfig is_struct
  }
}
```

349
CT.CLOUDFRONT.PR.11 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
    CachePolicy:
        Type: AWS::CloudFront::CachePolicy
        Properties:
            CachePolicyConfig:
                DefaultTTL: 20
                MaxTTL: 20
                MinTTL: 19
            Name:
                Fn::Sub: ${AWS::StackName}-example-cache-policy
        ParametersInCacheKeyAndForwardedToOrigin: null
        CookiesConfig:
            CookieBehavior: none
            EnableAcceptEncodingGzip: false
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
CachePolicy:
  Type: AWS::CloudFront::CachePolicy
  Properties:
    CachePolicyConfig:
      DefaultTTL: 20
      MaxTTL: 20
      MinTTL: 19
    Name:
      Fn::Sub: ${AWS::StackName}-example-cache-policy
ParametersInCacheKeyAndForwardedToOrigin:
  CookiesConfig:
    CookieBehavior: none
  EnableAcceptEncodingGzip: false
  HeadersConfig:
    HeaderBehavior: none
  QueryStringsConfig:
    QueryStringBehavior: none
CloudFrontDistribution:
  Type: AWS::CloudFront::Distribution
  Properties:
    DistributionConfig:
      Enabled: false
      DefaultCacheBehavior:
        ViewerProtocolPolicy: https-only
        TargetOriginId: exampleOrigin
        CachePolicyId:
          Ref: CachePolicy
    Origins:
      - Id: exampleOrigin
        DomainName: example.com
        CustomOriginConfig:
          OriginProtocolPolicy: https-only
          OriginSSLProtocols:
            - SSLv3
AWS CloudTrail controls

Topics

- [CT.CLOUDTRAIL.PR.1] Require an AWS CloudTrail trail to have encryption at rest activated (p. 352)
- [CT.CLOUDTRAIL.PR.2] Require an AWS CloudTrail trail to have log file validation activated (p. 358)
- [CT.CLOUDTRAIL.PR.3] Require an AWS CloudTrail trail to have an Amazon CloudWatch Logs log group configuration (p. 363)
- [CT.CLOUDTRAIL.PR.4] Require an AWS CloudTrail Lake event data store to enable encryption at rest with an AWS KMS key (p. 369)

[CT.CLOUDTRAIL.PR.1] Require an AWS CloudTrail trail to have encryption at rest activated

This control checks whether your AWS CloudTrail is configured to use the server-side encryption (SSE) AWS KMS key encryption.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CloudTrail::Trail
- **AWS CloudFormation guard rule:** [CT.CLOUDTRAIL.PR.1 rule specification (p. 353)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CLOUDTRAIL.PR.1 rule specification (p. 353)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CLOUDTRAIL.PR.1 example templates (p. 355)]

Explanation

For an added layer of security for your sensitive CloudTrail log files, you should use server-side encryption with AWS KMS keys (SSE-KMS) for your CloudTrail log files for encryption at rest. Note that by default, the log files delivered by CloudTrail to your buckets are encrypted by Amazon server-side encryption with Amazon S3-managed encryption keys (SSE-S3).

Remediation for rule failure

Set the KMSKeyId property to a valid KMS key.

The examples that follow show how to implement this remediation.

AWS CloudTrail trail - Example

AWS CloudTrail Trail configured to use server-side encryption with AWS KMS keys (SSE-KMS). The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "CloudTrail": {
      "Type": "AWS::CloudTrail::Trail",
      "Properties": {
```
YAML example

CloudTrail:
  Type: AWS::CloudTrail::Trail
  Properties:
    IsLogging: true
    KMSKeyId: !Ref 'KMSKey'
    S3BucketName: !Ref 'LoggingBucket'

CT.CLOUDTRAIL.PR.1 rule specification

```plaintext
# ###################################
##       Rule Specification        ##
####################################
#
# Rule Identifier:
#   cloud_trail_encryption_enabled_check
#
# Description:
#   This rule checks whether AWS CloudTrail is configured to use the server-side encryption (SSE) AWS KMS key encryption.
#
# Reports on:
#   AWS::CloudTrail::Trail
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#     And: The input document does not contain any AWS CloudTrail trails
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#     And: The input document contains a CloudTrail trail resource
#     And: 'KMSKeyId' is not present
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#     And: The input document contains a CloudTrail trail resource
#     And: 'KMSKeyId' has been provided and is set to an empty string or a non-valid local reference to a KMS key or
#         Alias
```
# Constants

let CLOUDTRAIL_TRAIL_TYPE = "AWS::CloudTrail::Trail"
let INPUT_DOCUMENT = this

# Assignments

let cloudtrail_trails = Resources.*[ Type == %CLOUDTRAIL_TRAIL_TYPE ]

# Primary Rules

rule cloud_trail_encryption_enabled_check when is_cfn_template(%INPUT_DOCUMENT) {
  %cloudtrail_trails not empty {
    check_cloudtrail_kms_key_configuration(%cloudtrail_trails.Properties)
    <<
      [CT.CLOUDTRAIL.PR.1]: Require an AWS CloudTrail trail to have encryption at rest
      activated
      [FIX]: Set the 'KMSKeyId' property to a valid KMS key.
    >>
  }
}

rule cloud_trail_encryption_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDTRAIL_TRAIL_TYPE) {
  check_cloudtrail_kms_key_configuration(%INPUT_DOCUMENT, %CLOUDTRAIL_TRAIL_TYPE.resourceProperties)
  <<
    [CT.CLOUDTRAIL.PR.1]: Require an AWS CloudTrail trail to have encryption at rest
    activated
    [FIX]: Set the 'KMSKeyId' property to a valid KMS key.
  >>
}

# Parameterized Rules

rule check_cloudtrail_kms_key_configuration(cloudtrail_trail) {
  %cloudtrail_trail {
    # Scenario 2
    KMSKeyId exists
    # Scenario 3 and 4
    check_is_string_and_not_empty(KMSKeyId) or check_kms_key_id_local_ref(KMSKeyId)
  }
}

rule check_kms_key_id_local_ref(key_ref) {
  %key_ref {
    check_local_references(%INPUT_DOCUMENT, this, "AWS::KMS::Key") or check_local_references(%INPUT_DOCUMENT, this, "AWS::KMS::Alias")
  }
}

# Utility Rules


CT.CLOUDTRAIL.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
KMSKey:
  Type: AWS::KMS::Key
  Properties:
    KeyPolicy:
      Version: 2012-10-17
      Id: example-cloudtrail-key-policy
      Statement:
        - Sid: Enable IAM User Permissions
          Effect: Allow
          Principal:
            AWS:
              Fn::Sub: arn:${AWS::Partition}:iam::${AWS::AccountId}:root
              Action: kms:*
              Resource: '*'
        - Sid: Allow CloudTrail to encrypt logs
          Effect: Allow
**Proactive controls**

```
Action: "kms:GenerateDataKey"
Principal:
  Service: "cloudtrail.amazonaws.com"
Resource: '*'
Condition:
  StringLike:
    "kms:EncryptionContext:aws:cloudtrail:arn": [
      Fn::Sub: "arn:aws:cloudtrail:*:${AWS::AccountId}:trail/*"
    ]
  StringEquals:
    "aws:SourceArn":
      Fn::Sub: "arn:aws:cloudtrail:${AWS::Region}:${AWS::AccountId}:trail/
${AWS::StackName}-example-trail"

- Sid: Allow CloudTrail to describe key
  Effect: Allow
  Principal:
    Service: "cloudtrail.amazonaws.com"
  Action: kms:DescribeKey
  Resource: '*'

- Sid: Allow principals in the account to decrypt log files
  Effect: Allow
  Principal:
    AWS: "*
  Action:
    - "kms:Decrypt"
    - "kms:ReEncryptFrom"
  Resource: '*'
  Condition:
    StringEquals:
      "kms:CallerAccount":
        Ref: AWS::AccountId
      "kms:EncryptionContext:aws:cloudtrail:arn":
        Fn::Sub: "arn:aws:cloudtrail:*:${AWS::AccountId}:trail/*"

LoggingBucket:
  Type: AWS::S3::Bucket
LoggingBucketPolicy:
  Type: AWS::S3::BucketPolicy
Properties:
  Bucket:
    Ref: LoggingBucket
  PolicyDocument:
    Version: 2012-10-17
    Statement:
      - Action:
        - 's3:GetBucketAcl'
        Effect: Allow
        Resource:
          Fn::Join:
            - ''
            - - 'arn:aws:s3:::'
            - Ref: LoggingBucket
        Principal:
          Service: "cloudtrail.amazonaws.com"
        Condition:
          StringEquals:
            "aws:SourceArn":
              Fn::Sub: "arn:aws:cloudtrail:${AWS::Region}:${AWS::AccountId}:trail/
${AWS::StackName}-example-trail"
      - Action:
        - 's3:PutObject'
        Effect: Allow
        Resource:
          Fn::Join:
            - ''
            - - 'arn:aws:s3:::'
            - Ref: LoggingBucket
```

356
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
LoggingBucket:
  Type: AWS::S3::Bucket
LoggingBucketPolicy:
  Type: AWS::S3::BucketPolicy
Properties:
  Bucket:
    Ref: LoggingBucket
  PolicyDocument:
    Version: 2012-10-17
    Statement:
      Action:
      - 's3:GetBucketAcl'
      Effect: Allow
      Resource:
        Fn::Join:
        - ''
        - - 'arn:aws:s3:::
        - Ref: LoggingBucket
    Principal:
      Service: "cloudtrail.amazonaws.com"
    Condition:
      StringEquals:
        "aws:SourceArn":
        Fn::Sub: "arn:aws:cloudtrail:${AWS::Region}:${AWS::AccountId}:trail/
${AWS::StackName}-example-trail"
      Action:
      - 's3:PutObject'
      Effect: Allow
      Resource:
        Fn::Join:
        - ''
        - - 'arn:aws:s3:::
        - Ref: LoggingBucket
        - /AWSLogs/
        - Ref: AWS::AccountId
        - /*
    Principal:
[CT.CLOUDTRAIL.PR.2] Require an AWS CloudTrail trail to have log file validation activated

This control checks whether log file integrity validation is enabled on an AWS CloudTrail trail.

- **Control objective:** Manage secrets
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CloudTrail::Trail
- **AWS CloudFormation guard rule:** [CT.CLOUDTRAIL.PR.2 rule specification](p. 359)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CLOUDTRAIL.PR.2 rule specification](p. 359)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CLOUDTRAIL.PR.2 example templates](p. 361)

Explanation

CloudTrail log file validation creates a digitally-signed digest file that contains a hash of each log that CloudTrail writes to Amazon S3. You can use these digest files to determine whether a log file was changed, deleted, or unchanged after CloudTrail delivered the log.

AWS Control Tower recommends that you enable file validation on all trails. Log file validation provides additional integrity checks of CloudTrail logs.

Remediation for rule failure

Set the CloudTrail resource EnableLogFileValidation property to true.

The examples that follow show how to implement this remediation.

**AWS CloudTrail trail - Example**

AWS CloudTrail trail configured with log file validation. The example is shown in JSON and in YAML.

**JSON example**

```json
Service: "cloudtrail.amazonaws.com"
Condition:
  StringEquals:
    's3:x-amz-acl': 'bucket-owner-full-control'
    "aws:SourceArn":
      Fn::Sub: "arn:aws:cloudtrail:${AWS::Region}:${AWS::AccountId}:trail/
${AWS::StackName}-example-trail"
CloudTrail:
  Type: AWS::CloudTrail::Trail
  Properties:
    IsLogging: true
    TrailName:
      Fn::Sub: ${AWS::StackName}-example-trail
    S3BucketName:
      Ref: LoggingBucket
```
[ "CloudTrail": {
   "Type": "AWS::CloudTrail::Trail",
   "Properties": {
      "IsLogging": true,
      "S3BucketName": {
         "Ref": "LoggingBucket"
      },
      "KMSKeyId": {
         "Ref": "KMSKey"
      },
      "EnableLogFileValidation": true
   }
}]

YAML example

CloudTrail:
  Type: AWS::CloudTrail::Trail
  Properties:
    IsLogging: true
    S3BucketName: !Ref 'LoggingBucket'
    KMSKeyId: !Ref 'KMSKey'
    EnableLogFileValidation: true

CT.CLOUDTRAIL.PR.2 rule specification

# #####################################################################
##       Rule Specification        
# #####################################################################
#
# Rule Identifier:  
#    cloud_trail_log_file_validation_enabled_check  
#
# Description:  
#    This control checks whether log file integrity validation is enabled on an AWS  
#    CloudTrail trail.  
#
# Reports on:  
#    AWS::CloudTrail::Trail  
#
# Evaluates:  
#    AWS CloudFormation, AWS CloudFormation hook  
#
# Rule Parameters:  
#    None  
#
# Scenarios:  
#    Scenario: 1  
#      Given: The input document is an AWS CloudFormation or CloudFormation hook document  
#      And: The input document does not contain any CloudTrail trails  
#      Then: SKIP  
#    Scenario: 2  
#      Given: The input document is an AWS CloudFormation or CloudFormation hook document  
#      And: The input document contains a CloudTrail trail resource  
#      And: 'EnableLogFileValidation' is not present  
#      Then: FAIL
# Scenario: 3
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains a CloudTrail trail resource
And: 'EnableLogFileValidation' is present and is set to a value other than bool(true)
Then: FAIL
# Scenario: 4
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains a CloudTrail trail resource
And: 'EnableLogFileValidation' is present and is set to bool(true)
Then: PASS

# Constants
let CLOUDTRAIL_TRAIL_TYPE = "AWS::CloudTrail::Trail"
let INPUT_DOCUMENT = this

# Assignments
let cloudtrail_trails = Resources.*[ Type == %CLOUDTRAIL_TRAIL_TYPE ]

# Primary Rules
rule cloud_trail_log_file_validation_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%cloudtrail_trails not empty {
  check(%cloudtrail_trails.Properties)
  <<
  [CT.CLOUDTRAIL.PR.2]: Require an AWS CloudTrail trail to have log file validation
  activated
  [FIX]: Set the CloudTrail resource 'EnableLogFileValidation' property to true.
  >>
}

rule cloud_trail_log_file_validation_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDTRAIL_TRAIL_TYPE) {
  check(%INPUT_DOCUMENT.%CLOUDTRAIL_TRAIL_TYPE.resourceProperties)
  <<
  [CT.CLOUDTRAIL.PR.2]: Require an AWS CloudTrail trail to have log file validation
  activated
  [FIX]: Set the CloudTrail resource 'EnableLogFileValidation' property to true.
  >>
}

# Parameterized Rules
rule check(cloudtrail_trail){
  %cloudtrail_trail {
    # Scenario 2
    EnableLogFileValidation exists
    # Scenario 3 and 4
    EnableLogFileValidation == true
  }
}

# Utility Rules
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
CT.CLOUDTRAIL.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  LoggingBucket:
    Type: AWS::S3::Bucket
  LoggingBucketPolicy:
    Type: AWS::S3::BucketPolicy
    Properties:
      Bucket:
        Ref: LoggingBucket
      PolicyDocument:
        Version: 2012-10-17
        Statement:
        - Action:
        - 's3:GetBucketAcl'
        Effect: Allow
        Resource:
          Fn::Join:
          - ''
        - 'arn:aws:s3:::'
        - Ref: LoggingBucket
      Principal:
        Service: "cloudtrail.amazonaws.com"
        Condition:
          StringEquals:
            "aws:SourceArn":
            Fn::Sub: "arn:aws:cloudtrail:${AWS::Region}:${AWS::AccountId}:trail/${AWS::StackName}-example-trail"
        - Action:
        - 's3:PutObject'
        Effect: Allow
        Resource:
          Fn::Join:
          - ''
        - 'arn:aws:s3:::'
        - Ref: LoggingBucket
        - /AWSLogs/
        - Ref: AWS::AccountId
        - /*
      Principal:
        Service: "cloudtrail.amazonaws.com"
        Condition:
          StringEquals:
            's3:x-amz-acl': 'bucket-owner-full-control'
            "aws:SourceArn":
            Fn::Sub: "arn:aws:cloudtrail:${AWS::Region}:${AWS::AccountId}:trail/${AWS::StackName}-example-trail"
  CloudTrail:
    Type: AWS::CloudTrail::Trail
    Properties:
      IsLogging: true
```
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
LoggingBucket:
  Type: AWS::S3::Bucket
LoggingBucketPolicy:
  Type: AWS::S3::BucketPolicy
Properties:
  Bucket:
    Ref: LoggingBucket
  PolicyDocument:
    Version: 2012-10-17
    Statement:
    - Action:
      - 's3:GetBucketAcl'
      Effect: Allow
      Resource:
        Fn::Join:
        - '
          - 'arn:aws:s3:::'
          - Ref: LoggingBucket
    Principal:
      Service: "cloudtrail.amazonaws.com"
    Condition:
      StringEquals:
        "aws:SourceArn":
        Fn::Sub: "arn:aws:cloudtrail:${AWS::Region}:${AWS::AccountId}:trail/
        ${AWS::StackName}-example-trail"
    - Action:
      - 's3:PutObject'
      Effect: Allow
      Resource:
        Fn::Join:
        - '
          - 'arn:aws:s3:::'
          - Ref: LoggingBucket
          - /AWSLogs/
        - Ref: AWS::AccountId
        - /*
      Principal:
        Service: "cloudtrail.amazonaws.com"
    Condition:
      StringEquals:
        "s3:x-amz-acl": 'bucket-owner-full-control'
        "aws:SourceArn":
        Fn::Sub: "arn:aws:cloudtrail:${AWS::Region}:${AWS::AccountId}:trail/
        ${AWS::StackName}-example-trail"
CloudTrail:
  Type: AWS::CloudTrail::Trail
  Properties:
    IsLogging: true
    TrailName:
      Fn::Sub: ${AWS::StackName}-example-trail
    S3BucketName:
      Ref: LoggingBucket
    EnableLogFileValidation: false
[CT.CLOUDTRAIL.PR.3] Require an AWS CloudTrail trail to have an Amazon CloudWatch Logs log group configuration

This control checks whether your AWS CloudTrail trail is configured to send logs to Amazon CloudWatch Logs.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CloudTrail::Trail
- **AWS CloudFormation guard rule:** CT.CLOUDTRAIL.PR.3 rule specification (p. 364)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.CLOUDTRAIL.PR.3 rule specification (p. 364)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.CLOUDTRAIL.PR.3 example templates (p. 367)

**Explanation**

CloudTrail records AWS API calls that are made in a given account. The recorded information includes: the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters and the response elements returned by the AWS service.

CloudTrail uses Amazon S3 for log file storage and delivery. You can capture CloudTrail logs in a specified S3 bucket for long-term analysis. To perform real-time analysis, you can configure CloudTrail to send logs to CloudWatch Logs.

For a trail that is enabled in all AWS Regions in an account, CloudTrail sends log files from all of those Regions to a CloudWatch Logs log group.

AWS CloudTrail recommends that you send CloudTrail logs to CloudWatch Logs. Note that this recommendation is intended to ensure that account activity is captured, monitored, and appropriately alarmed on. You can use CloudWatch Logs to set this up with your AWS services. This recommendation does not preclude the use of a different solution.

Sending CloudTrail logs to CloudWatch Logs facilitates real-time and historic activity logging based on user, API, resource, and IP address. You can use this approach to establish alarms and notifications for anomalous or sensitivity account activity.

**Remediation for rule failure**

Set the CloudWatchLogsLogGroupArn and CloudWatchLogsRoleArn properties.

The examples that follow show how to implement this remediation.

**AWS CloudTrail trail - Example**

AWS CloudTrail trail configured to send events to Amazon CloudWatch Logs. The example is shown in JSON and in YAML.

**JSON example**
Proactive controls

YAML example

CloudTrail:
  Type: AWS::CloudTrail::Trail
  Properties:
    IsLogging: true
    S3BucketName: !Ref 'LoggingBucket'
    CloudWatchLogsRoleArn: !GetAtt 'LogRole.Arn'
    CloudWatchLogsLogGroupArn: !GetAtt 'LogGroup.Arn'

CT.CLOUDTRAIL.PR.3 rule specification

# ###################################################################
## Rule Specification       #
# ###################################################################
# Rule Identifier:
# cloud_trail_cloud_watch_logs_enabled_check
# Description:
# This rule checks whether AWS CloudTrail trails are configured to send logs to Amazon CloudWatch Logs.
# Reports on:
# AWS::CloudTrail::Trail
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
# None
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document does not contain any AWS CloudTrail trails
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an AWS CloudTrail trail resource
# And: 'CloudWatchLogsLogGroupArn' or 'CloudWatchLogsRoleArn' is not present
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an AWS CloudTrail trail resource
# And: 'CloudWatchLogsLogGroupArn' is set to a non-empty string or a valid local reference to a log group
# And: 'CloudWatchLogsRoleArn' is set to an empty string or a non-valid local reference
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an AWS CloudTrail trail resource
# And: 'CloudWatchLogsLogGroupArn' is set to an empty string or an invalid local reference
# And: 'CloudWatchLogsRoleArn' is set to a non-empty string or a valid local reference to an IAM role
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an AWS CloudTrail trail resource
# And: 'CloudWatchLogsLogGroupArn' is set to a non-empty string or a valid local reference to a log group
# And: 'CloudWatchLogsRoleArn' is set to a non-empty string or a valid local reference
# Then: PASS

# Constants
let CLOUDTRAIL_TRAIL_TYPE = "AWS::CloudTrail::Trail"
let INPUT_DOCUMENT = this

# Assignments
let cloudtrail_trails = Resources.*[ Type == %CLOUDTRAIL_TRAIL_TYPE ]

# Primary Rules
# rule cloud_trail_cloud_watch_logs_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
#   %cloudtrail_trails not empty {
#     check_cloudtrail_log_group_configuration(%cloudtrail_trails.Properties)
#     <<
#     [CT.CLOUDTRAIL.PR.3]: Require an AWS CloudTrail trail to have an CloudTrail log group configuration
#     [FIX]: Set the 'CloudWatchLogsLogGroupArn' and 'CloudWatchLogsRoleArn' properties.
#     >>
#   }
# rule cloud_trail_cloud_watch_logs_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDTRAIL_TRAIL_TYPE) {
#   check_cloudtrail_log_group_configuration(%INPUT_DOCUMENT, %CLOUDTRAIL_TRAIL_TYPE.resourceProperties)
#   <<
#   [CT.CLOUDTRAIL.PR.3]: Require an AWS CloudTrail trail to have an CloudTrail log group configuration
#   [FIX]: Set the 'CloudWatchLogsLogGroupArn' and 'CloudWatchLogsRoleArn' properties.
#   >>
#}
# Parameterized Rules

```csharp
rule check_cloudtrail_log_group_configuration(cloudtrail_trail) {
  %cloudtrail_trail {
    # Scenario 2
    CloudWatchLogsLogGroupArn exists
    CloudWatchLogsRoleArn exists

    # Scenario 3, 4 and 5
    check_cloudwatch_log_group_arn(CloudWatchLogsLogGroupArn)
    check_cloudwatch_log_role_arn(CloudWatchLogsRoleArn)
  }
}
```

```csharp
rule check_cloudwatch_log_group_arn(log_group) {
  %log_group {
    check_is_string_and_not_empty(this) or
    check_local_references(%INPUT_DOCUMENT, this, "AWS::Logs::LogGroup")
  }
}
```

```csharp
rule check_cloudwatch_log_role_arn(log_role) {
  %log_role {
    check_is_string_and_not_empty(this) or
    check_local_references(%INPUT_DOCUMENT, this, "AWS::IAM::Role")
  }
}
```

# Utility Rules

```csharp
rule check_is_string_and_not_empty(value) {
  %value {
    this is_string
    this !~ \A[^\s]*\z/
  }
}
```

```csharp
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
```

```csharp
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

```csharp
rule check_local_references(doc, reference_properties, referenced RESOURCE_TYPE) {
  %reference_properties {
    'Fn::GetAtt' {
      query_for_resource(%doc, this[0], %referenced RESOURCE_TYPE)
      <<Local Stack reference was invalid>>
    } or Ref {
      query_for_resource(%doc, this, %referenced RESOURCE_TYPE)
      <<Local Stack reference was invalid>>
    }
  }
}
```

```csharp
rule query_for_resource(doc, resource_key, referenced RESOURCE_TYPE) {

```
CT.CLOUDTRAIL.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  LoggingBucket:
    Type: AWS::S3::Bucket
  LoggingBucketPolicy:
    Type: AWS::S3::BucketPolicy
  Properties:
    Bucket:
      Ref: LoggingBucket
    PolicyDocument:
      Version: 2012-10-17
      Statement:
        - Action:
          - 's3:GetBucketAcl'
          Effect: Allow
          Resource:
            Fn::Join:
            - ''
            - - 'arn:aws:s3:::'
            - Ref: LoggingBucket
            Principal:
              Service: "cloudtrail.amazonaws.com"
            Condition:
              StringEquals:
                "aws:SourceArn":
                Fn::Sub: "arn:aws:cloudtrail:${AWS::Region}:${AWS::AccountId}:trail/
$(AWS::StackName)-example-trail"
                - Action:
                  - 's3:PutObject'
                  Effect: Allow
                  Resource:
                    Fn::Join:
                    - ''
                    - - 'arn:aws:s3:::'
                    - Ref: LoggingBucket
                    - /AWSLogs/
                    - Ref: AWS::AccountId
                    - /*
                    Principal:
                      Service: "cloudtrail.amazonaws.com"
                    Condition:
                      StringEquals:
                        's3:x-amz-acl': 'bucket-owner-full-control'
                        "aws:SourceArn":
                        Fn::Sub: "arn:aws:cloudtrail:${AWS::Region}:${AWS::AccountId}:trail/
$(AWS::StackName)-example-trail"
  CloudWatchLogsRole:
    Type: "AWS::IAM::Role"
    Properties:
AssumeRolePolicyDocument:
  Version: '2012-10-17'
  Statement:
  - Sid: AssumeRole
    Effect: Allow
    Principal:
      Service: 'cloudtrail.amazonaws.com'
    Action: 'sts:AssumeRole'
  Policies:
  - PolicyName: 'cloudtrail-policy'
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
      - Effect: Allow
        Action:
        - 'logs:CreateLogStream'
        - 'logs:PutLogEvents'
        Resource:
          Fn::GetAtt: [LogGroup, Arn]

LogGroup:
  Type: AWS::Logs::LogGroup
  Properties: {}

CloudTrail:
  Type: AWS::CloudTrail::Trail
  Properties:
    IsLogging: true
    TrailName:
      Fn::Sub: ${AWS::StackName}-example-trail
    S3BucketName:
      Ref: LoggingBucket
    CloudWatchLogsRoleArn:
      Fn::GetAtt:
        - CloudWatchLogsRole
        - Arn
    CloudWatchLogsLogGroupArn:
      Fn::GetAtt:
        - LogGroup
        - Arn

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  LoggingBucket:
    Type: AWS::S3::Bucket
  LoggingBucketPolicy:
    Type: AWS::S3::BucketPolicy
    Properties:
      Bucket:
        Ref: LoggingBucket
      PolicyDocument:
        Version: 2012-10-17
        Statement:
        - Action:
          - 's3:GetBucketAcl'
          Effect: Allow
          Resource:
            Fn::Join:
              - ''
              - - 'arn:aws:s3:::'
              - - Ref: LoggingBucket
            Principal:
              Service: "cloudtrail.amazonaws.com"
[CT.CLOUDTRAIL.PR.4] Require an AWS CloudTrail Lake event data store to enable encryption at rest with an AWS KMS key

This control checks whether a CloudTrail Lake event data store is encrypted at rest with a KMS key.

- **Control objective**: Encrypt data at rest
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::CloudTrail::EventDataStore
- **AWS CloudFormation guard rule**: CT.CLOUDTRAIL.PR.4 rule specification (p. 370)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.CLOUDTRAIL.PR.4 rule specification (p. 370)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.CLOUDTRAIL.PR.4 example templates (p. 373)

Explanation

Encrypting data at rest reduces the risk that a user not authenticated to AWS may obtain access to data stored on disk. For added control over encryption keys, you can use customer-managed keys from AWS KMS. You have full control over these KMS keys. You can establish and maintain their key policies, IAM
policies, and grants, enable and disable the keys, rotate their cryptographic material, add tags, create aliases that refer to the KMS keys, and schedule the KMS keys for deletion.

**Usage considerations**

- All events in an AWS CloudTrail Lake event data store are encrypted by CloudTrail using a KMS key that AWS owns and manages for you. For added control over encryption keys, you can use customer-managed keys from AWS KMS. For more information, see [AWS KMS Concepts](#) in the [AWS KMS Developer Guide](#).

**Remediation for rule failure**

Set the `KmsKeyId` parameter to the ARN of an AWS KMS customer-managed key, configured with permissions that allow the CloudTrail service principal to use the key.

The examples that follow show how to implement this remediation.

**CloudTrail Lake event data store - Example**

CloudTrail Lake event data store configured to encrypt data at rest with an AWS KMS key. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "CloudTrailEventDataStore": {
    "Type": "AWS::CloudTrail::EventDataStore",
    "Properties": {
      "Name": {
        "Fn::Sub": "${AWS::StackName}-example"
      },
      "TerminationProtectionEnabled": false,
      "KmsKeyId": {
        "Fn::GetAtt": [
          "KMSKey",
          "Arn"
        ]
      }
    }
  }
}
```

**YAML example**

```yaml
CloudTrailEventDataStore:
  Type: AWS::CloudTrail::EventDataStore
  Properties:
    Name: !Sub '${AWS::StackName}-example'
    TerminationProtectionEnabled: false
    KmsKeyId: !GetAtt 'KMSKey.Arn'
```

**CT.CLOUDTRAIL.PR.4 rule specification**

---

370
# #################################################################
##       Rule Specification        ##
# #################################################################
#
# Rule Identifier:
#   cloud_trail_event_datastore_encrypted_at_rest_kms_check
#
# Description:
#   This control checks whether a CloudTrail Lake event data store is encrypted at rest with a KMS key.
#
# Reports on:
#   AWS::CloudTrail::EventDataStore
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any CloudTrail event data store resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a CloudTrail event data store resource
#     And: 'KmsKeyId' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a CloudTrail event data store resource
#     And: 'KmsKeyId' has been provided as an empty string or invalid local reference to a KMS keyID or alias
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation Hook Document
#     And: The input document contains a CloudTrail event data store resource
#     And: 'KmsKeyId' has been provided as a non-empty string or valid local reference to a KMS keyID or alias
#     Then: PASS
#
# Constants
#
let CLOUDTRAIL_EVENT_DATASTORE_TYPE = "AWS::CloudTrail::EventDataStore"
let INPUT_DOCUMENT = this
#
# Assignments
#
let cloudata_event_datastores = Resources.*[ Type == %CLOUDTRAIL_EVENT_DATASTORE_TYPE ]
#
# Primary Rules
#
rule cloud_trail_event_datastore_encrypted_at_rest_kms_check when
   is_cfn_template(%INPUT_DOCUMENT)
   %cloudtrail_event_datastores not empty {
      check(%cloudtrail_event_datastores.Properties)
[CT.CLOUDTRAIL.PR.4]: Require an CloudTrail Lake event data store to enable encryption at rest with an AWS KMS key

[FIX]: Set the 'KmsKeyId' parameter to the ARN of an AWS KMS customer-managed key, configured with permissions that allow the CloudTrail service principal to use the key.

rule cloud_trail_event_datastore_encrypted_at_rest_kms_check when
is_cfn_hook(%INPUT_DOCUMENT, %CLOUDTRAIL_EVENT_DATASTORE_TYPE) {
  check(%INPUT_DOCUMENT.%CLOUDTRAIL_EVENT_DATASTORE_TYPE.resourceProperties)

  [CT.CLOUDTRAIL.PR.4]: Require an CloudTrail Lake event data store to enable encryption at rest with an AWS KMS key

  [FIX]: Set the 'KmsKeyId' parameter to the ARN of an AWS KMS customer-managed key, configured with permissions that allow the CloudTrail service principal to use the key.

}
rule query_for_resource(doc, resource_key, referenced_RESOURCE_TYPE) {
    let referenced_resource = %doc.Resources[ keys == %resource_key ]
    %referenced_resource not empty
    %referenced_resource {
        Type == %referenced_RESOURCE_TYPE
    }
}

**CT.CLOUDTRAIL.PR.4 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```
Resources:
    KMSKey:
      Type: AWS::KMS::Key
      Properties:
        KeyPolicy:
          Version: 2012-10-17
          Id: example-policy
          Statement:
            - Sid: Enable IAM User Permissions
              Effect: Allow
              Principal:
                AWS:
                  Fn::Sub: arn:${AWS::Partition}:iam::${AWS::AccountId}:root
                Action: kms:*
                Resource: '*'
            - Sid: Allow CloudTrail to encrypt event data store
              Effect: Allow
              Principal:
                Service: "cloudtrail.amazonaws.com"
                Action:
                  - "kms:GenerateDataKey"
                  - "kms:Decrypt"
                Resource: "*"
              KeySpec: SYMMETRIC_DEFAULT
              EnableKeyRotation: true
    CloudTrailEventDataStore:
      Type: AWS::CloudTrail::EventDataStore
      Properties:
        Name:
          Fn::Sub: ${AWS::StackName}-example
        TerminationProtectionEnabled: false
        MultiRegionEnabled: false
        KmsKeyId:
          Fn::GetAtt:
            - KMSKey
            - Arn
```

**FAIL Example** - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
    CloudTrailEventDataStore:
```

373
Type: AWS::CloudTrail::EventDataStore
Properties:
  Name:
    Fn::Sub: ${AWS::StackName}-example
  TerminationProtectionEnabled: false
  MultiRegionEnabled: false

Amazon CloudWatch controls

Topics
- [CT.CLOUDWATCH.PR.1] Require an Amazon CloudWatch alarm to have an action configured for the alarm state (p. 374)
- [CT.CLOUDWATCH.PR.2] Require an Amazon CloudWatch log group to be retained for at least one year (p. 378)
- [CT.CLOUDWATCH.PR.3] Require an Amazon CloudWatch log group to be encrypted at rest with an AWS KMS key (p. 381)
- [CT.CLOUDWATCH.PR.4] Require an Amazon CloudWatch alarm to have actions activated (p. 386)

[CT.CLOUDWATCH.PR.1] Require an Amazon CloudWatch alarm to have an action configured for the alarm state

This control checks whether an Amazon CloudWatch alarm has at least one action configured for the alarm state.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CloudWatch::Alarm
- **AWS CloudFormation guard rule:** [CT.CLOUDWATCH.PR.1 rule specification](p. 375)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CLOUDWATCH.PR.1 rule specification](p. 375)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.CLOUDWATCH.PR.1 example templates](p. 377)

Explanation

AWS Control Tower recommends configuring actions for alarms to alert you automatically when an alarm is in the alarm state and the monitored metric is outside the defined threshold. This configuration ensures that alarms are monitored, and that necessary actions are taken when the alarm is triggered. Monitoring alarms help you identify unusual activities and respond quickly to security and operational issues. You can specify the actions an alarm should take when it goes into OK, ALARM, and INSUFFICIENT_DATA states. The most common CloudWatch alarm action in the alarm state is to notify one or more users by sending a message to an Amazon Simple Notification Service (Amazon SNS) topic.

Remediation for rule failure

Set AlarmActions to a list with one or more alarm action values.
The examples that follow show how to implement this remediation.

**Amazon CloudWatch Alarm - Example**

An Amazon CloudWatch alarm configured to notify an SNS topic when the CloudWatch alarm is in the alarm state. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "Alarm": {
      "Type": "AWS::CloudWatch::Alarm",
      "Properties": {
         "ComparisonOperator": "GreaterThanOrEqualToThreshold",
         "EvaluationPeriods": 1,
         "Period": 300,
         "Threshold": 1.0,
         "Namespace": "AWS/Lambda",
         "MetricName": "Errors",
         "TreatMissingData": "missing",
         "Statistic": "Sum",
         "DatapointsToAlarm": 1,
         "ActionsEnabled": true,
         "AlarmActions": [ {
            "Ref": "Topic"
         } ]
      }
   }
}
```

**YAML example**

```
Alarm:
   Type: AWS::CloudWatch::Alarm
   Properties:
      ComparisonOperator: GreaterThanOrEqualToThreshold
      EvaluationPeriods: 1
      Period: 300
      Threshold: 1.0
      Namespace: AWS/Lambda
      MetricName: Errors
      TreatMissingData: missing
      Statistic: Sum
      DatapointsToAlarm: 1
      ActionsEnabled: true
      AlarmActions:
         - !Ref 'Topic'
```

**CT.CLOUDWATCH.PR.1 rule specification**

```
# ###################################
##       Rule Specification        ##
#   * Rule Specification *
#  *
#  *
#  *
#  *
#  *
#  *
#  *
#  *
```

375
# Rule Identifier:
# cloudwatch_alarm_action_check
#
# Description:
# This control checks whether an Amazon CloudWatch alarm has at least one action
# configured for the alarm state.
#
# Reports on:
# AWS::CloudWatch::Alarm
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#   document
#   And: The input document does not contain any CloudWatch alarm resources
#   Then: SKIP
# Scenario: 2
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#   document
#   And: The input document contains a CloudWatch alarm resource
#   And: 'AlarmActions' has not been provided
#   Then: FAIL
# Scenario: 3
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#   document
#   And: The input document contains a CloudWatch alarm resource
#   And: 'AlarmActions' has been provided as an empty list
#   Then: FAIL
# Scenario: 4
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation Hook
#   Document
#   And: The input document contains a CloudWatch alarm resource
#   And: 'AlarmActions' has been provided as a non-empty list
#   Then: PASS
#
# Constants
#
# let CLOUDWATCH_ALARM_TYPE = "AWS::CloudWatch::Alarm"
let INPUT_DOCUMENT = this
#
# Assignments
#
let cloudwatch_alarms = Resources.*[ Type == %CLOUDWATCH_ALARM_TYPE ]
#
# Primary Rules
#
rule cloudwatch_alarm_action_check when is_cfn_template(%INPUT_DOCUMENT)
    %cloudwatch_alarms not empty {  
    check(%cloudwatch_alarms.Properties)
    <<
      [CT.CLOUDWATCH.PR.1]: Require an Amazon CloudWatch alarm to have an action
      configured for the alarm state
      [FIX]: Set 'AlarmActions' to a list with one or more alarm action values.
    >>
}
rule cloudwatch_alarm_action_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDWATCH_ALARM_TYPE) {
    check(%INPUT_DOCUMENT.%CLOUDWATCH_ALARM_TYPE.resourceProperties)
    <<
    [CT.CLOUDWATCH.PR.1]: Require an Amazon CloudWatch alarm to have an action configured for the alarm state
    [FIX]: Set 'AlarmActions' to a list with one or more alarm action values.
    >>
}

# Parameterized Rules
#
rule check(cloudwatch_alarm){
    %cloudwatch_alarm {
        # Scenario 2
        AlarmActions exists
        # Scenarios 3 and 4
        AlarmActions is_list
        AlarmActions not empty
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.CLOUDWATCH.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  Topic:
    Type: AWS::SNS::Topic
    Properties: {}
  Alarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
        ComparisonOperator: GreaterThanOrEqualToThreshold
        EvaluationPeriods: 1
        Period: 300
        Threshold: 1.0
        Namespace: AWS/Lambda
        MetricName: Errors
        TreatMissingData: missing
        Statistic: Sum
        DatapointsToAlarm: 1
        ActionsEnabled: true
        AlarmActions:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
Alarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    ComparisonOperator: GreaterThanOrEqualToThreshold
    EvaluationPeriods: 1
    Period: 300
    Threshold: 1.0
    Namespace: AWS/Lambda
    MetricName: Errors
    TreatMissingData: missing
    Statistic: Sum
    DatapointsToAlarm: 1
    ActionsEnabled: true

[CT.CLOUDWATCH.PR.2] Require an Amazon CloudWatch log group to be retained for at least one year

This control checks whether an Amazon CloudWatch Log Group retention period is set to a value greater than or equal to 365 days.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Logs::LogGroup
- **AWS CloudFormation guard rule:** [CT.CLOUDWATCH.PR.2 rule specification](p. 379)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CLOUDWATCH.PR.2 rule specification](p. 379)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.CLOUDWATCH.PR.2 example templates](p. 381)

Explanation

Amazon CloudWatch Logs centralizes the logs from all of your systems, applications, and AWS services in a single, highly scalable service. You can use Amazon CloudWatch Logs to monitor, store, and retrieve your log files from Amazon EC2 instances, CloudTrail, Route 53, and other sources. Retaining your logs for at least one year can help you comply with log retention standards.

Remediation for rule failure

Omit the field value of RetentionInDays to adopt the default retention setting of Never expire, or set RetentionInDays to an integer value greater than or equal to 365.

The examples that follow show how to implement this remediation.
Amazon CloudWatch Log Group - Example

An Amazon CloudWatch log group configured to retain logs for one year (365 days). The example is shown in JSON and in YAML.

**JSON example**

```
{
    "LogGroup": {
        "Type": "AWS::Logs::LogGroup",
        "Properties": {
            "RetentionInDays": 365
        }
    }
}
```

**YAML example**

```
LogGroup:
    Type: AWS::Logs::LogGroup
    Properties:
        RetentionInDays: 365
```

**CT.CLOUDWATCH.PR.2 rule specification**

```
# ###################################
##       Rule Specification        ##
####################################
#
# Rule Identifier:
#   cloudwatch_log_group_retention_period_check
#
# Description:
#   This control checks whether an Amazon CloudWatch Log Group retention period is set to a value greater than or equal to 365 days.
#
# Reports on:
#   AWS::Logs::LogGroup
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#           And: The input document does not contain any CloudWatch log group resources
#           Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#           And: The input document contains a CloudWatch log group resource
#           And: 'RetentionInDays' has been provided and set to a non integer value or
```
integer value less than 365
Then: FAIL

Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document contains a CloudWatch log group resource
And: 'RetentionInDays' has not been provided
Then: PASS

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation Hook
document
And: The input document contains a CloudWatch log group resource
And: 'RetentionInDays' has been provided and set to an integer value greater than
or equal to 365
Then: PASS

Constants

let CLOUDWATCH_LOGS_TYPE = "AWS::Logs::LogGroup"
let MINIMUM_RETENTION_IN_DAYS = 365
let INPUT_DOCUMENT = this

Assignments

let cloudwatch_log_groups = Resources.*[ Type == %CLOUDWATCH_LOGS_TYPE ]

Primary Rules

rule cloudwatch_log_group_retention_period_check when is_cfn_template(%INPUT_DOCUMENT)
%cloudwatch_log_groups not empty {
    check(%cloudwatch_log_groups.Properties)
    [CT.CLOUDWATCH.PR.2]: Require an Amazon CloudWatch log group to be retained for at
    least one year
    [FIX]: Omit the field value of 'RetentionInDays' to adopt the default retention
    setting of 'Never expire', or set 'RetentionInDays' to an integer value greater than or
    equal to 365.
    }
}

rule cloudwatch_log_group_retention_period_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDWATCH_LOGS_TYPE) {
    check(%INPUT_DOCUMENT.%CLOUDWATCH_LOGS_TYPE.resourceProperties)
    [CT.CLOUDWATCH.PR.2]: Require an Amazon CloudWatch log group to be retained for at
    least one year
    [FIX]: Omit the field value of 'RetentionInDays' to adopt the default retention
    setting of 'Never expire', or set 'RetentionInDays' to an integer value greater than or
    equal to 365.
    }
}

Parameterized Rules

rule check(cloudwatch_log_group){
    %cloudwatch_log_group {
        # Scenario 3
        RetentionInDays not exists or

        # Scenarios 2 and 4
        RetentionInDays >= %MINIMUM_RETENTION_IN_DAYS
    }
}
CT.CLOUDWATCH.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  LogGroup:
    Type: AWS::Logs::LogGroup
    Properties:
      RetentionInDays: 365
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
  LogGroup:
    Type: AWS::Logs::LogGroup
    Properties:
      RetentionInDays: 1
```

[CT.CLOUDWATCH.PR.3] Require an Amazon CloudWatch log group to be encrypted at rest with an AWS KMS key

This control checks whether an Amazon CloudWatch Logs log group is encrypted at rest with an AWS KMS key

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Logs::LogGroup
- **AWS CloudFormation guard rule:** CT.CLOUDWATCH.PR.3 rule specification (p. 382)

Details and examples
For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the:
CT.CLOUDWATCH.PR.3 rule specification (p. 382)

For examples of PASS and FAIL CloudFormation templates related to this control, see:
CT.CLOUDWATCH.PR.3 example templates (p. 385)

Explanation

Amazon CloudWatch Logs log groups are encrypted by default using server-side encryption. For added control over encryption keys, you can use customer-managed keys from AWS KMS. You have full control over these KMS keys, including establishing and maintaining their key policies, IAM policies, and grants, enabling and disabling the keys, rotating their cryptographic material, adding tags, creating aliases that refer to the KMS keys, and scheduling the KMS keys for deletion.

Remediation for rule failure

Set KmsKeyId to the ARN of an AWS KMS customer-managed key configured with permissions that allow the CloudWatch service principal to use the key.

The examples that follow show how to implement this remediation.

Amazon CloudWatch Logs Group - Example

An Amazon CloudWatch log group configured to encrypt logs with an AWS KMS customer-managed key. The example is shown in JSON and in YAML.

JSON example

```json
{
    "LogGroup": {
        "Type": "AWS::Logs::LogGroup",
        "Properties": {
            "KmsKeyId": {
                "Fn::GetAtt": [
                    "KMSKey",
                    "Arn"
                ]
            }
        }
    }
}
```

YAML example

```yaml
LogGroup:
    Type: AWS::Logs::LogGroup
    Properties:
        KmsKeyId: !GetAtt 'KMSKey.Arn'
```

CT.CLOUDWATCH.PR.3 rule specification

```yaml
# ###################################
```
## Rule Specification

### Rule Identifier:
- `cloudwatch_log_group_encrypted_check`

### Description:
- This control checks whether an Amazon CloudWatch log group is encrypted at rest with an AWS KMS key.

### Reports on:
- `AWS::Logs::LogGroup`

### Evaluates:
- AWS CloudFormation, AWS CloudFormation hook

### Rule Parameters:
- None

### Scenarios:

#### Scenario: 1
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document does not contain any CloudWatch log group resources
  - Then: SKIP

#### Scenario: 2
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document contains a CloudWatch log group resource
  - And: 'KmsKeyId' has not been provided
  - Then: FAIL

#### Scenario: 3
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document contains a CloudWatch log group resource
  - And: 'KmsKeyId' has been provided as an empty string or invalid local reference to a KMS Key
  - Then: FAIL

#### Scenario: 4
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document contains a CloudWatch log group resource
  - And: 'KmsKeyId' has been provided as a non-empty string or valid local reference to a KMS Key
  - Then: PASS

### Constants

- `CLOUDWATCH_LOGS_TYPE = "AWS::Logs::LogGroup"`
- `INPUT_DOCUMENT = this`

### Assignments

- `let cloudwatch_log_groups = Resources.*[ Type == %CLOUDWATCH_LOGS_TYPE ]`

### Primary Rules

- `rule cloudwatch_log_group_encrypted_check when is_cfn_template(%INPUT_DOCUMENT) %cloudwatch_log_groups not empty {`
  
  `check(%cloudwatch_log_groups.Properties)`

  `<<`

  `[CT.CLOUDWATCH.PR.3]: Require an Amazon CloudWatch log group to be encrypted at rest with an AWS KMS key`
[FIX]: Set 'KmsKeyId' to the ARN of an AWS KMS customer managed key configured with permissions that allow the CloudWatch service principal to use the key.

rule cloudwatch_log_group_encrypted_check when is_cfn_hook(%INPUT_DOCUMENT, %CLOUDWATCH_LOGS_TYPE) {
  check(%INPUT_DOCUMENT.%CLOUDWATCH_LOGS_TYPE.resourceProperties)
  <![CT.CLOUDWATCH.PR.3]: Require an Amazon CloudWatch log group to be encrypted at rest with an AWS KMS key
  [FIX]: Set 'KmsKeyId' to the ARN of an AWS KMS customer managed key configured with permissions that allow the CloudWatch service principal to use the key.]
}

# Parameterized Rules
#
rule check(cloudwatch_log_group){
  %cloudwatch_log_group {
    # Scenario 2
    KmsKeyId exists
    # Scenario 3 and 4
    check_is_string_and_not_empty(KmsKeyId) or
    check_local_references(%INPUT_DOCUMENT, KmsKeyId, "AWS::KMS::Key")
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

rule check_is_string_and_not_empty(value) {
  %value {
    this is_string
    this != /\A\s*\z/
  }
}

rule check_local_references(doc, reference_properties, referenced_resource_type) {
  %reference_properties {
    'Fn::GetAtt' {
      query_for_resource(%doc, this[0], %referenced_resource_type)
      <Local Stack reference was invalid>>
    } or Ref {
      query_for_resource(%doc, this, %referenced_resource_type)
      <Local Stack reference was invalid>>
    }
  }
}

rule query_for_resource(doc, resource_key, resource_type) {
  let referenced_resource = %doc.Resources[ keys == %resource_key ]
  %referenced_resource not empty
  %referenced_resource {
Type == %resource_type

CT.CLOUDWATCH.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
KMSKey:
  Type: AWS::KMS::Key
  Properties:
    KeyPolicy:
      Version: 2012-10-17
      Id: example-cloudwatch-logs-key-policy
      Statement:
        - Sid: Enable IAM User Permissions
          Effect: Allow
          Principal:
            AWS:
              Fn::Sub: arn:${AWS::Partition}:iam::${AWS::AccountId}:root
          Action: kms:*
          Resource: "*"
        - Sid: Enable Logs
          Effect: Allow
          Principal:
            Service:
              Fn::Sub: logs.${AWS::Region}.amazonaws.com
          Action:
            - kms:Encrypt*
            - kms:Decrypt*
            - kms:ReEncrypt*
            - kms:GenerateDataKey*
            - kms:Describe
            Resource: "*"
          Condition:
            ArnEquals:
              kms:EncryptionContext:aws:logs:arn:
              Fn::Sub: arn:${AWS::Partition}:logs:${AWS::Region}:${AWS::AccountId}:*
LogGroup:
  Type: AWS::Logs::LogGroup
  Properties:
    KmsKeyId:
      Fn::GetAtt:
        - KMSKey
        - Arn

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
LogGroup:
  Type: AWS::Logs::LogGroup
  Properties: { }
[CT.CLOUDWATCH.PR.4] Require an Amazon CloudWatch alarm to have actions activated

This control checks whether an Amazon CloudWatch alarm has actions enabled.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CloudWatch::Alarm
- **AWS CloudFormation guard rule:**  [CT.CLOUDWATCH.PR.4 rule specification (p. 387)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the:  [CT.CLOUDWATCH.PR.4 rule specification (p. 387)]
- For examples of PASS and FAIL CloudFormation templates related to this control, see:  [CT.CLOUDWATCH.PR.4 example templates (p. 389)]

**Explanation**

Alarm actions automatically alert you when a monitored metric is outside the defined threshold. If the alarm action is deactivated, no actions are executed when the alarm changes state, so you won't be alerted to changes in monitored metrics. AWS Control Tower recommends activating CloudWatch alarm actions to help you respond quickly to security and operational issues.

**Remediation for rule failure**

Set `ActionsEnabled` to `true` or do not provide the `ActionsEnabled` property.

The examples that follow show how to implement this remediation.

**Amazon CloudWatch Alarm - Example**

An Amazon CloudWatch alarm configured with alarm actions enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "Alarm": {
        "Type": "AWS::CloudWatch::Alarm",
        "Properties": {
            "AlarmActions": [
                {
                    "Ref": "Topic"
                }
            ],
            "ComparisonOperator": "GreaterThanOrEqualToThreshold",
            "EvaluationPeriods": 1,
            "Period": 300,
            "Threshold": 1.0,
            "Namespace": "AWS/Lambda",
            "MetricName": "Errors",
            "TreatMissingData": "missing",
            "Statistic": "Sum",
            "DatapointsToAlarm": 1,
            "ActionsEnabled": true
        }
    }
}
```

386
YAML example

Alarm:
  Type: AWS::CloudWatch::Alarm
  Properties:
    AlarmActions:
    - !Ref 'Topic'
    ComparisonOperator: GreaterThanOrEqualToThreshold
    EvaluationPeriods: 1
    Period: 300
    Threshold: 1.0
    Namespace: AWS/Lambda
    MetricName: Errors
    TreatMissingData: missing
    Statistic: Sum
    DatapointsToAlarm: 1
    ActionsEnabled: true

CT.CLOUDWATCH.PR.4 rule specification

# ####################################################################
#      Rule Specification      #
# ####################################################################
#
# Rule Identifier:
#   cloudwatch_alarm_action_enabled_check
#
# Description:
#   This control checks whether an Amazon CloudWatch alarm has actions enabled.
#
# Reports on:
#   AWS::CloudWatch::Alarm
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any CloudWatch alarm resources
#     Then: SKIP
#
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a CloudWatch alarm resource
#     And: 'ActionsEnabled' has been provided and set to a value other than bool(true)
#     Then: FAIL
#
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a CloudWatch alarm resource
#
# And: 'ActionsEnabled' has not been provided
# Then: PASS

# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation Hook Document
# And: The input document contains a CloudWatch alarm resource
# And: 'ActionsEnabled' has been provided with a value of bool(true)
# Then: PASS

# Constants
#
let CLOUDWATCH_ALARM_TYPE = "AWS::CloudWatch::Alarm"
let INPUT_DOCUMENT = this
#
# Assignments
#
let cloudwatch_alarms = Resources.*[ Type == %CLOUDWATCH_ALARM_TYPE ]
#
# Primary Rules
#
rule cloudwatch_alarm_action_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%cloudwatch_alarms not empty {
  check(%cloudwatch_alarms.Properties)
  %cloudwatch_alarms not empty {
    <<
    [CT.CLOUDWATCH.PR.4]: Require an Amazon CloudWatch alarm to have actions activated
    [FIX]: Set 'ActionsEnabled' to 'true' or do not provide the 'ActionsEnabled'
    property.
    >>
  }
}

rule cloudwatch_alarm_action_enabled_check when is_cfn_hook(%INPUT_DOCUMENT,
%CLOUDWATCH_ALARM_TYPE) {
  check(%INPUT_DOCUMENT.%CLOUDWATCH_ALARM_TYPE.resourceProperties)
  %INPUT_DOCUMENT.%CLOUDWATCH_ALARM_TYPE.resourceProperties {
    <<
    [CT.CLOUDWATCH.PR.4]: Require an Amazon CloudWatch alarm to have actions activated
    [FIX]: Set 'ActionsEnabled' to 'true' or do not provide the 'ActionsEnabled'
    property.
    >>
  }
}

# Parameterized Rules
#
rule check(cloudwatch_alarm){
  %cloudwatch_alarm {
    # Scenario 3
    ActionsEnabled not exists or
    # Scenarios 2 and 4
    ActionsEnabled == true
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
%doc.%RESOURCE_TYPE.resourceProperties exists

CT.CLOUDWATCH.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  Topic:
    Type: AWS::SNS::Topic
    Properties: {}
  Alarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmActions:
        - Ref: Topic
      ComparisonOperator: GreaterThanOrEqualToThreshold
      EvaluationPeriods: 1
      Period: 300
      Threshold: 1.0
      Namespace: AWS/Lambda
      MetricName: Errors
      TreatMissingData: missing
      Statistic: Sum
      DatapointsToAlarm: 1
      ActionsEnabled: true
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```yaml
Resources:
  Topic:
    Type: AWS::SNS::Topic
    Properties: {}
  Alarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmActions:
        - Ref: Topic
      ComparisonOperator: GreaterThanOrEqualToThreshold
      EvaluationPeriods: 1
      Period: 300
      Threshold: 1.0
      Namespace: AWS/Lambda
      MetricName: Errors
      TreatMissingData: missing
      Statistic: Sum
      DatapointsToAlarm: 1
      ActionsEnabled: false
```

AWS CodeBuild controls

Topics
[CT.CODEBUILD.PR.1] Require OAuth on GitHub or Bitbucket source repository URLs for AWS CodeBuild projects (p. 390)

[CT.CODEBUILD.PR.2] Require any AWS CodeBuild project environment variable to encrypt credentials in environment variables (p. 398)

[CT.CODEBUILD.PR.3] Require any AWS CodeBuild project environment to have logging configured (p. 404)

[CT.CODEBUILD.PR.4] Require any AWS CodeBuild project to deactivate privileged mode when running (p. 412)

[CT.CODEBUILD.PR.5] Require encryption on all AWS CodeBuild project artifacts (p. 418)

[CT.CODEBUILD.PR.6] Require encryption on all Amazon S3 logs for AWS CodeBuild projects (p. 427)

[CT.CODEBUILD.PR.1] Require OAuth on GitHub or Bitbucket source repository URLs for AWS CodeBuild projects

This control checks whether the GitHub or Bitbucket source repository URL contains either personal access tokens or a user name and password.

- **Control objective:** Use strong authentication
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CodeBuild::Project
- **AWS CloudFormation guard rule:** [CT.CODEBUILD.PR.1 rule specification (p. 393)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CODEBUILD.PR.1 rule specification (p. 393)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CODEBUILD.PR.1 example templates (p. 396)]

Explanation

Authentication credentials should never be stored or transmitted in clear text or appear in the repository URL. Instead of personal access tokens or username and password, you should use OAuth to grant authorization for accessing GitHub or Bitbucket repositories. Using personal access tokens or a username and password could expose your credentials to unintended data exposure and unauthorized access.

Usage considerations

- This control applies only to AWS CodeBuild projects with a primary or secondary source type of GitHub or Bitbucket.

Remediation for rule failure

Remove any embedded credentials from repository URLs in AWS CodeBuild project source configurations. Instead, connect your CodeBuild projects to GitHub or Bitbucket repositories by configuring GitHub Access Token or Bitbucket App Password credentials in the AWS Management Console or AWS CLI.

The examples that follow show how to implement this remediation.

390
AWS CodeBuild Project - Example One

AWS CodeBuild project configured with a GitHub primary source location that does not contain a personal access token. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "CodeBuildProject": {
        "Type": "AWS::CodeBuild::Project",
        "Properties": {
            "Artifacts": {
                "Type": "NO_ARTIFACTS"
            },
            "Environment": {
                "ComputeType": "BUILD_GENERAL1_SMALL",
                "Image": "aws/codebuild/standard:4.0",
                "Type": "LINUX_CONTAINER"
            },
            "ServiceRole": {
                "Fn::GetAtt": [
                    "CodeBuildServiceRole",
                    "Arn"
                ]
            },
            "Source": {
                "BuildSpec": "version: 0.2
phases:
  install:
    commands:
      - npm install
  build:
    commands:
      - npm test
artifacts:
  files:
    - '**/*'
Type": "GITHUB",
"Location": "https://github.com/username/repo.git"
    }
} }
```

**YAML example**

```yaml
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:4.0
      Type: LINUX_CONTAINER
    ServiceRole: !GetAtt 'CodeBuildServiceRole.Arn'
  Source:
    BuildSpec: |
    version: 0.2
    phases:
      install:
        commands:
          - npm install
      build:
        commands:
          - npm test
        artifacts:
          files:
            - '**/*'
            Type: GITHUB
```

391
Location: https://github.com/username/repo.git

The examples that follow show how to implement this remediation.

**AWS CodeBuild Project - Example Two**

AWS CodeBuild project configured with primary and secondary source locations that do not contain credentials or personal access tokens. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "CodeBuildProject": {
    "Type": "AWS::CodeBuild::Project",
    "Properties": {
      "Artifacts": {
        "Type": "NO_ARTIFACTS"
      },
      "Environment": {
        "ComputeType": "BUILD_GENERAL1_SMALL",
        "Image": "aws/codebuild/standard:4.0",
        "Type": "LINUX_CONTAINER"
      },
      "ServiceRole": {
        "Fn::GetAtt": [
          "CodeBuildServiceRole",
          "Arn"
        ]
      },
      "Source": {
        "BuildSpec": "version: 0.2
install
build:
commands:
- npm install
- npm test
artifacts:
files:
- '**/*'",
"Type": "BITBUCKET",
"Location": "https://bitbucket.org/user/repo.git"
      },
      "SecondarySources": [
        {
          "Type": "GITHUB",
          "Location": "https://github.com/username/repo.git",
          "SourceIdentifier": "GitHubSource"
        }
      ]
    }
  }
}
```

**YAML example**

```yaml
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:4.0
      Type: LINUX_CONTAINER
    ServiceRole: !GetAtt 'CodeBuildServiceRole.Arn'
```

---

392
Source:
   BuildSpec: |
    version: 0.2
   phases:
    install:
       commands:
        - npm install
   build:
       commands:
        - npm test
   artifacts:
    files:
     - '**/*'
Type: BITBUCKET
Location: https://bitbucket.org/user/repo.git
SecondarySources:
 - Type: GITHUB
  Location: https://github.com/username/repo.git
SourceIdentifier: GitHubSource

CT.CODEBUILD.PR.1 rule specification

# #############################################################################
##       Rule Specification        
# #############################################################################
#
# Rule Identifier:
#   codebuild_project_source_repo_url_check
#
# Description:
#   This control checks whether the GitHub or Bitbucket source repository URL contains either personal access tokens or a username and password.
#
# Reports on:
#   AWS::CodeBuild::Project
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any CodeBuild project resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a CodeBuild project resource
#     And: 'Source' configuration is not of 'Type' 'GITHUB' or 'BITBUCKET'
#     And: 'SecondarySources' configuration is not provided or is provided and does not have any item of 'Type'
#      'GITHUB' or 'BITBUCKET'
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a CodeBuild project resource
And: 'Source' configuration is of 'Type' 'GITHUB' or 'BITBUCKET'
And: 'Source' configuration has a 'Location' that contains credentials (username and password for BitBucket and Access Token for GitHub)
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CodeBuild project resource
And: 'SecondarySources' configuration is provided
And: 'SecondarySources' configuration has one or more items of 'Type' 'GITHUB' or 'BITBUCKET'
And: 'SecondarySources' configuration has one or more items with 'Location' that contains credentials (username and password for BitBucket and Access Token for GitHub)
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CodeBuild project resource
And: 'Source' configuration is of 'Type' 'GITHUB' or 'BITBUCKET'
And: 'Source' configuration has a 'Location' that does not contain credentials (username and password for BitBucket and Access Token for GitHub)
And: 'SecondarySources' configuration is not provided or is provided and does not have any item of 'Type' 'GITHUB' or 'BITBUCKET'
Then: PASS

Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CodeBuild project resource
And: 'Source' configuration is of 'Type' 'GITHUB' or 'BITBUCKET'
And: 'Source' configuration has a 'Location' that does not contain credentials (username and password for BitBucket and Access Token for GitHub)
And: 'SecondarySources' configuration is provided
And: 'SecondarySources' configuration has one or more items of 'Type' 'GITHUB' or 'BITBUCKET'
And: 'SecondarySources' configuration has one or more items with 'Location' that does not contain credentials (username and password for BitBucket and Access Token for GitHub)
Then: PASS

# Constants
let CODEBUILD_PROJECT_TYPE = "AWS::CodeBuild::Project"

# Assignments
let codebuild_project = Resources.*[ Type == %CODEBUILD_PROJECT_TYPE ]

# Primary Rules
rule codebuild_project_source_repo_url_check when is_cfn_template(%INPUT_DOCUMENT) %codebuild_project not empty {
  check(%codebuild_project.Properties)
<<

394
[CT.CODEBUILD.PR.1]: Require OAuth on GitHub or Bitbucket source repository URLs for AWS CodeBuild projects

[FIX]: Remove any embedded credentials from repository URLs in AWS CodeBuild project source configurations. Instead, connect your CodeBuild projects to 'GitHub' or 'Bitbucket' repositories by configuring 'GitHub Access Token' or 'Bitbucket App Password' credentials in the AWS Management Console or AWS CLI.

```python
rule codebuild_project_source_repo_url_check when is_cfn_hook(%INPUT_DOCUMENT, %CODEBUILD_PROJECT_TYPE) {
    check(%INPUT_DOCUMENT.%CODEBUILD_PROJECT_TYPE.resourceProperties)

    [CT.CODEBUILD.PR.1]: Require OAuth on GitHub or Bitbucket source repository URLs for AWS CodeBuild projects
    [FIX]: Remove any embedded credentials from repository URLs in AWS CodeBuild project source configurations. Instead, connect your CodeBuild projects to 'GitHub' or 'Bitbucket' repositories by configuring 'GitHub Access Token' or 'Bitbucket App Password' credentials in the AWS Management Console or AWS CLI.
}
```

## Parameterized Rules

```python
rule check(codebuild_project) {
    %codebuild_project[
        filter_github_or_bitbucket_source_configuration(this) or
        filter_github_or_bitbucket_secondary_sources_configuration(this)
    ] {
        # Scenario 3, 5 and 6
        check_source(Source)
        # Scenario 4 and 6
        check_secondary_sources(this)
    }
}
```

```python
rule filter_github_or_bitbucket_source_configuration(codebuild_project) {
    %codebuild_project {
        Source exists
        Source is_struct
        Source {
            Type == "GITHUB" or
            Type == "BITBUCKET"
        }
    }
}
```

```python
rule filter_github_or_bitbucket_secondary_sources_configuration(codebuild_project) {
    %codebuild_project {
        SecondarySources exists
        SecondarySources is_list
        SecondarySources not empty

        some SecondarySources[*] {
            Type == "GITHUB" or
            Type == "BITBUCKET"
        }
    }
}
```

```python
rule check_source(codebuild_source) {
    %codebuild_source [
        Type == "GITHUB"
    ] {
        Location exists
    }
}
Location == %GITHUB_COMPLIANT_URL_PATTERN

%codebuild_source [
  Type == "BITBUCKET"
] {
  Location exists
  Location == %BITBUCKET_COMPLIANT_URL_PATTERN
}

rule check_secondary_sources(codebuild_project) {
  %codebuild_project [
    # Scenario 2
    SecondarySources exists
    SecondarySources is_list
    SecondarySources not empty
  ] {
    # Scenario 4 and 6
    SecondarySources[*] {
      check_source(this)
    }
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.CODEBUILD.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
CodeBuildServiceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: codebuild.amazonaws.com
            Action: sts:AssumeRole
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
CodeBuildServiceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: codebuild.amazonaws.com
          Action: sts:AssumeRole
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:4.0
      Type: LINUX_CONTAINER
    ServiceRole:
      Fn::GetAtt:
        - CodeBuildServiceRole
        - Arn
    Source:
      BuildSpec: |
        version: 0.2
        phases:
          install:
            commands:
              - npm install
          build:
            commands:
              - npm test
        artifacts:
          files:
            - '***/*'
        Type: GITHUB
        Location: https://github.com/username/repo.git
[CT.CODEBUILD.PR.2] Require any AWS CodeBuild project environment variable to encrypt credentials in environment variables

This control checks whether AWS CodeBuild projects contain environment variables AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY stored as PLAINTEXT.

- **Control objective:** Use strong authentication
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CodeBuild::Project
- **AWS CloudFormation guard rule:** [CT.CODEBUILD.PR.2 rule specification](p. 400)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CODEBUILD.PR.2 rule specification](p. 400)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CODEBUILD.PR.2 example templates](p. 402)

**Explanation**

Authentication credentials AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY should never be stored in clear text, as this could lead to unintended data exposure and unauthorized access.

**Usage considerations**

- This control only applies to AWS CodeBuild projects configured with AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY environment variables

**Remediation for rule failure**

Use PARAMETER_STORE or SECRETS_MANAGER to store values for environment variables named AWS_ACCESS_KEY_ID or AWS_SECRET_ACCESS_KEY.

The examples that follow show how to implement this remediation.

**AWS CodeBuild Project - Example**

AWS CodeBuild project configured to use credentials stored in AWS Secrets Manager. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "CodeBuildProject": {
        "Type": "AWS::CodeBuild::Project",
        "Properties": {
            "Artifacts": {
                "Type": "NO_ARTIFACTS"
            }
        }
    }
}
```
YAML example

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
Properties:
  Artifacts:
    Type: NO_ARTIFACTS
  Environment:
    ComputeType: BUILD_GENERAL1_SMALL
    Image: aws/codebuild/standard:4.0
    Type: LINUX_CONTAINER
    EnvironmentVariables:
      - Name: AWS_ACCESS_KEY_ID
        Type: SECRETS_MANAGER
        Value: sample_secret:access_key_id
      - Name: AWS_SECRET_ACCESS_KEY
        Type: SECRETS_MANAGER
        Value: sample_secret:secret_access_key
  ServiceRole: !GetAtt 'CodeBuildServiceRole.Arn'
  Source:
    Type: NO_SOURCE
    BuildSpec: |
      version: 0.2
      phases:
        install:
          commands:
            - npm install
        build:
          commands:
            - npm test
      artifacts:
        files: '**/*.js'
```

CT.CODEBUILD.PR.2 rule specification

```plaintext
# ###################################################################
## Rule Specification     ##
# ###################################################################
#
# Rule Identifier:
# codebuild_project_envvar_awscred_check
#
# Description:
# This control checks whether AWS CodeBuild projects contain environment variables
# 'AWS_ACCESS_KEY_ID' and 'AWS_SECRET_ACCESS_KEY' stored as 'PLAINTEXT'.
#
# Reports on:
# AWS::CodeBuild::Project
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation Hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document does not contain any CodeBuild project resources
# Then: SKIP
#
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a CodeBuild project resource
# And: 'Environment' configuration does not contains 'EnvironmentVariables'
# Then: SKIP
#
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a CodeBuild project resource
# And: 'Environment' configuration contains 'EnvironmentVariables'
# And: 'EnvironmentVariables' contain variables named 'AWS_ACCESS_KEY_ID' or
# 'AWS_SECRET_ACCESS_KEY'
# And: 'Type' is not provided for 'AWS_ACCESS_KEY_ID' and 'AWS_SECRET_ACCESS_KEY'
# environment variables or is
# provided as an empty string.
# Then: FAIL
#
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a CodeBuild project resource
# And: 'Environment' configuration contains 'EnvironmentVariables'
# And: 'EnvironmentVariables' contain variables named 'AWS_ACCESS_KEY_ID' or
# 'AWS_SECRET_ACCESS_KEY'
# And: 'Type' is set to 'PLAINTEXT' for 'AWS_ACCESS_KEY_ID' or
# 'AWS_SECRET_ACCESS_KEY' environment variables
# Then: FAIL
#
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
```
And: The input document contains a CodeBuild project resource
And: 'Environment' configuration contains 'EnvironmentVariables'
And: 'EnvironmentVariables' does not contain variables named 'AWS_ACCESS_KEY_ID' or 'AWS_SECRET_ACCESS_KEY'
Then: PASS

Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document contains a CodeBuild project resource
And: 'Environment' configuration contains 'EnvironmentVariables'
And: 'EnvironmentVariables' contain variables named 'AWS_ACCESS_KEY_ID' or 'AWS_SECRET_ACCESS_KEY'
And: 'Type' is provided as a non-empty string and not set to 'PLAINTEXT' for 'AWS_ACCESS_KEY_ID' or 'AWS_SECRET_ACCESS_KEY' environment variables
Then: PASS

# Constants
let CODEBUILD_PROJECT_TYPE = "AWS::CodeBuild::Project"
let AWS_CREDENTIAL_ENV_VAR_NAMES = [ "AWS_ACCESS_KEY_ID", "AWS_SECRET_ACCESS_KEY" ]
let INPUT_DOCUMENT = this

# Assignments
let codebuild_project = Resources.*[ Type == %CODEBUILD_PROJECT_TYPE ]

# Primary Rules
rule codebuild_project_envvar_awscred_check when is_cfn_template(%INPUT_DOCUMENT)
%codebuild_project not empty {
  check(%codebuild_project.Properties)
  %codebuild_project not empty {
    [CT.CODEBUILD.PR.2]: Require any AWS CodeBuild project environment variable to encrypt credentials in environment variables
    [FIX]: Use 'PARAMETER_STORE' or 'SECRETS_MANAGER' to store values for environment variables named 'AWS_ACCESS_KEY_ID' or 'AWS_SECRET_ACCESS_KEY'.
  }
}

rule codebuild_project_envvar_awscred_check when is_cfn_hook(%INPUT_DOCUMENT, %CODEBUILD_PROJECT_TYPE)
%INPUT_DOCUMENT.%CODEBUILD_PROJECT_TYPE.resourceProperties)
  check(%INPUT_DOCUMENT.%CODEBUILD_PROJECT_TYPE.resourceProperties)
  %INPUT_DOCUMENT.%CODEBUILD_PROJECT_TYPE.resourceProperties)
  [CT.CODEBUILD.PR.2]: Require any AWS CodeBuild project environment variable to encrypt credentials in environment variables
  [FIX]: Use 'PARAMETER_STORE' or 'SECRETS_MANAGER' to store values for environment variables named 'AWS_ACCESS_KEY_ID' or 'AWS_SECRET_ACCESS_KEY'.
}

# Parameterized Rules
rule check(codebuild_project) {
  %codebuild_project [ ]
  Scenario 2
  filter_codebuild_projects_with_environment_variables(this)
  Environment exists
  Environment is_struct
  Environment {
**CT.CODEBUILD.PR.2 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

**Resources:**
- CodeBuildServiceRole:
  - Type: AWS::IAM::Role
Properties:
  AssumeRolePolicyDocument:
    Version: '2012-10-17'
    Statement:
      - Effect: Allow
        Principal:
          Service: codebuild.amazonaws.com
        Action: sts:AssumeRole
CodeBuildProject:
  Type: AWS::CodeBuild::Project
Properties:
  Artifacts:
    Type: NO_ARTIFACTS
  Environment:
    ComputeType: BUILD_GENERAL1_SMALL
    Image: aws/codebuild/standard:4.0
    Type: LINUX_CONTAINER
    EnvironmentVariables:
      - Name: AWS_ACCESS_KEY_ID
        Type: SECRETS_MANAGER
        Value: example_secret:access_key_id
      - Name: AWS_SECRET_ACCESS_KEY
        Type: SECRETS_MANAGER
        Value: example_secret:secret_access_key
      - Name: some_other_variable
        Type: PLAINTEXT
        Value: example
  ServiceRole:
    Fn::GetAtt:
      - CodeBuildServiceRole
      - Arn
Source:
  Type: NO_SOURCE
BuildSpec: |
  version: 0.2
  phases:
    install:
      commands:
        - npm install
    build:
      commands:
        - npm test
  artifacts:
    files:
      - '**/*'

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  CodeBuildServiceRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: codebuild.amazonaws.com
            Action: sts:AssumeRole
  CodeBuildProject:
    Type: AWS::CodeBuild::Project
    Properties:
Artifacts:
  Type: NO_ARTIFACTS
Environment:
  ComputeType: BUILD_GENERAL1_SMALL
  Image: aws/codebuild/standard:4.0
  Type: LINUX_CONTAINER
EnvironmentVariables:
  - Name: AWS_ACCESS_KEY_ID
    Type: PLAINTEXT
    Value: EXAMPLE_ACCESS_KEY_ID
  - Name: AWS_SECRET_ACCESS_KEY
    Type: PLAINTEXT
    Value: EXAMPLE_SECRET_ACCESS_KEY
ServiceRole:
  Fn::GetAtt:
    - CodeBuildServiceRole
      - Arn
Source:
  Type: NO_SOURCE
BuildSpec: |
    version: 0.2
    phases:
      install:
        commands:
          - npm install
      build:
        commands:
          - npm test
      artifacts:
        files:
          - '**/*'

[CT.CODEBUILD.PR.3] Require any AWS CodeBuild project environment to have logging configured

This control checks whether AWS CodeBuild projects environment has at least one logging option enabled.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CodeBuild::Project
- **AWS CloudFormation guard rule:** [CT.CODEBUILD.PR.3 rule specification (p. 407)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CODEBUILD.PR.3 rule specification (p. 407)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CODEBUILD.PR.3 example templates (p. 411)]

Explanation

From a security perspective, logging is an important feature to enable, to assist future forensics efforts in case of a security incident. Correlating anomalies in CodeBuild projects with threat detections can increase confidence in the accuracy of those threat detections.
Remediation for rule failure

Set LogsConfig with a CloudWatchLogs or S3Logs configuration.

The examples that follow show how to implement this remediation.

AWS CodeBuild Project - Example One

AWS CodeBuild project configured to enable logging, by means of Amazon CloudWatch Logs. The example is shown in JSON and in YAML.

**JSON example**

```
{
  "CodeBuildProject": {
    "Type": "AWS::CodeBuild::Project",
    "Properties": {
      "Artifacts": {
        "Type": "NO_ARTIFACTS"
      },
      "Environment": {
        "ComputeType": "BUILD_GENERAL1_SMALL",
        "Image": "aws/codebuild/standard:4.0",
        "Type": "LINUX_CONTAINER"
      },
      "ServiceRole": {
        "Fn::GetAtt": [
          "CodeBuildServiceRole",
          "Arn"
        ]
      },
      "Source": {
        "Type": "NO_SOURCE",
        "BuildSpec": "version: 0.2
install
build
commands:

- npm install
- npm test"
      },
      "LogsConfig": {
        "CloudWatchLogs": {
          "Status": "ENABLED"
        }
      }
    }
  }
}
```

**YAML example**

```
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:4.0
      Type: LINUX_CONTAINER
    ServiceRole: !GetAtt 'CodeBuildServiceRole.Arn'
  Source:
    Type: NO_SOURCE
```
Proactive controls

BuildSpec: "version: 0.2
phases:
  install:
    commands:
      - npm install
  build:
    commands:
      - npm test"
LogsConfig:
  CloudWatchLogs:
    Status: ENABLED

The examples that follow show how to implement this remediation.

AWS CodeBuild Project - Example Two

AWS CodeBuild project configured to enable logging, by means of Amazon S3. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "CodeBuildProject": {
    "Type": "AWS::CodeBuild::Project",
    "Properties": {
      "Artifacts": {
        "Type": "NO_ARTIFACTS"
      },
      "Environment": {
        "ComputeType": "BUILD_GENERAL1_SMALL",
        "Image": "aws/codebuild/standard:4.0",
        "Type": "LINUX_CONTAINER"
      },
      "ServiceRole": {
        "Fn::GetAtt": [
          "CodeBuildServiceRole",
          "Arn"
        ]
      },
      "Source": {
        "Type": "NO_SOURCE",
        "BuildSpec": "version: 0.2
phases:
  install:
    commands:
      - npm install
  build:
    commands:
      - npm test"
      }},
      "LogsConfig": {
        "S3Logs": {
          "Status": "ENABLED",
          "Location": {
            "Fn::Join": [
              "/",
              [
                "Ref": "S3Bucket"
              ],
              "path/to/directory"
            ]
          }
        }
      }
    }
  }
}
```

**YAML example**

```yaml
BuildSpec: "version: 0.2
phases:
  install:
    commands:
      - npm install
  build:
    commands:
      - npm test"
LogsConfig:
  CloudWatchLogs:
    Status: ENABLED

The examples that follow show how to implement this remediation.

AWS CodeBuild Project - Example Two

AWS CodeBuild project configured to enable logging, by means of Amazon S3. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "CodeBuildProject": {
    "Type": "AWS::CodeBuild::Project",
    "Properties": {
      "Artifacts": {
        "Type": "NO_ARTIFACTS"
      },
      "Environment": {
        "ComputeType": "BUILD_GENERAL1_SMALL",
        "Image": "aws/codebuild/standard:4.0",
        "Type": "LINUX_CONTAINER"
      },
      "ServiceRole": {
        "Fn::GetAtt": [
          "CodeBuildServiceRole",
          "Arn"
        ]
      },
      "Source": {
        "Type": "NO_SOURCE",
        "BuildSpec": "version: 0.2
phases:
  install:
    commands:
      - npm install
  build:
    commands:
      - npm test"
      }},
      "LogsConfig": {
        "S3Logs": {
          "Status": "ENABLED",
          "Location": {
            "Fn::Join": [
              "/",
              [
                "Ref": "S3Bucket"
              ],
              "path/to/directory"
            ]
          }
        }
      }
    }
  }
}
```

**YAML example**
CodeBuildProject:
  Type: AWS::CodeBuild::Project
Properties:
  Artifacts:
    Type: NO_ARTIFACTS
Environment:
  ComputeType: BUILD_GENERAL1_SMALL
  Image: aws/codebuild/standard:4.0
    Type: LINUX_CONTAINER
  ServiceRole: !GetAtt 'CodeBuildServiceRole.Arn'
Source:
  Type: NO_SOURCE
  BuildSpec: "version: 0.2
    phases:
      install:
        commands:
          - npm install
      build:
        commands:
          - npm test"
LogsConfig:
  S3Logs:
    Status: ENABLED
    Location: !Join
      - /
      - - !Ref 'S3Bucket'
        - path/to/directory

CT.CODEBUILD.PR.3 rule specification

# #################################################################
## Rule Specification      ##
# #################################################################

# Rule Identifier:
#   codebuild_project_logging_enabled_check
# Description:
#   This control checks whether AWS CodeBuild projects environment has at least one logging
#   option enabled.
# Reports on:
#   AWS::CodeBuild::Project
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
#   None
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any CodeBuild project resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a CodeBuild project resource
#     And: 'LogsConfig' is not provided on the CodeBuild project resource
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a CodeBuild project resource
# Proactive controls

And: 'LogsConfig' is provided on the CodeBuild project resource
And: Neither 'CloudWatchLogs' or 'S3Logs' are present in 'LogsConfig'
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CodeBuild project resource
And: 'LogsConfig' is provided on the CodeBuild project resource
And: 'CloudWatchLogs' is not present in 'LogsConfig'
And: 'S3Logs' is present in 'LogsConfig' with 'Status' set to 'DISABLED'
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CodeBuild project resource
And: 'LogsConfig' is provided on the CodeBuild project resource
And: 'CloudWatchLogs' is not present in 'LogsConfig'
And: 'S3Logs' is present in 'LogsConfig' with 'Status' set to 'DISABLED'
Then: FAIL

Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CodeBuild project resource
And: 'LogsConfig' is provided on the CodeBuild project resource
And: 'CloudWatchLogs' and 'S3Logs' are present in 'LogsConfig' with 'Status' set to 'DISABLED'
Then: FAIL

Scenario: 7
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CodeBuild project resource
And: 'LogsConfig' is provided on the CodeBuild project resource
And: 'CloudWatchLogs' is not present in 'LogsConfig'
And: 'S3Logs' is present in 'LogsConfig' with 'Status' set to 'ENABLED'
And: 'Location' has not been provided in 'S3Logs', or has been provided as an empty string or invalid local reference
Then: FAIL

Scenario: 8
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CodeBuild project resource
And: 'LogsConfig' is provided on the CodeBuild project resource
And: 'CloudWatchLogs' is not present in 'LogsConfig'
And: 'S3Logs' is present in 'LogsConfig' with 'Status' set to 'ENABLED'
Then: PASS

Scenario: 9
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CodeBuild project resource
And: 'LogsConfig' is provided on the CodeBuild project resource
And: 'CloudWatchLogs' is not present in 'LogsConfig'
And: 'S3Logs' is present in 'LogsConfig' with 'Status' set to 'ENABLED'
And: 'Location' has been provided in 'S3Logs' as a non-empty string or valid local reference
Then: PASS

Scenario: 10
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a CodeBuild project resource
And: 'LogsConfig' is provided on the CodeBuild project resource
And: 'CloudWatchLogs' is present in 'LogsConfig' with 'Status' set to 'ENABLED'
And: 'S3Logs' is present in 'LogsConfig' with 'Status' set to 'ENABLED'
And: 'Location' has been provided in 'S3Logs' as a non-empty string or valid local reference
Then: PASS
# Constants
#
let CODEBUILD_PROJECT_TYPE = "AWS::CodeBuild::Project"
let INPUT_DOCUMENT = this

# Assignments
#
let codebuild_project = Resources.*[ Type == %CODEBUILD_PROJECT_TYPE ]

# Primary Rules
#
rule codebuild_project_logging_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
  %codebuild_project not empty {
    check(%codebuild_project.Properties) %codebuild_project not empty {
      [[CT.CODEBUILD.PR.3]: Require any AWS CodeBuild project environment to have logging configured
      [FIX]: Set 'LogsConfig' with a 'CloudWatchLogs' or 'S3Logs' configuration.
      }]

rule codebuild_project_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %CODEBUILD_PROJECT_TYPE) {
  check(%INPUT_DOCUMENT.%CODEBUILD_PROJECT_TYPE.resourceProperties) {
    [[CT.CODEBUILD.PR.3]: Require any AWS CodeBuild project environment to have logging configured
    [FIX]: Set 'LogsConfig' with a 'CloudWatchLogs' or 'S3Logs' configuration.
    }]
}

# Parameterized Rules
#
rule check(codebuild_project) {
  %codebuild_project {
    # Scenario 2
    LogsConfig exists
    LogsConfig is_struct

    LogsConfig {
      # Scenario 3
      check_cloudwatch_logs(this) or
      check_s3_logs(this)
    }
  }
}

rule check_cloudwatch_logs(codebuild_project) {
  %codebuild_project {
    # Scenario 4
    CloudWatchLogs exists
    CloudWatchLogs is_struct

    CloudWatchLogs {
      # Scenario 5, 6, 8 and 10
      Status exists
      Status == "ENABLED"
    }
  }
}
rule check_s3_logs(codebuild_project) {
    %codebuild_project {
        # Scenario 4
        S3Logs exists
        S3Logs is_struct
        S3Logs {
            # Scenario 4, 6, 9 and 10
            Status exists
            Status == "ENABLED"
            # Scenario 7, 9 and 10
            Location exists
            check_is_string_and_not_empty(Location) or
            check_local_references(%INPUT_DOCUMENT, Location, "AWS::S3::Bucket") or
            check_join_references(%INPUT_DOCUMENT, Location, "AWS::S3::Bucket"
        }
    }
}

rule check_join_references(doc, reference_properties, referenced_resource_type) {
    %reference_properties {
        'Fn::Join' {
            this is_list
            this not empty
            some this[1][*] { 
                check_local_references(%doc, this, %referenced_resource_type)
            }
        }
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

rule check_is_string_and_not_empty(value) {
    %value {
        this is_string
        this != /\A\s*\z/
    }
}

rule check_local_references(doc, reference_properties, referenced_resource_type) {
    %reference_properties {
        'Fn::GetAtt' {
            query_for_resource(%doc, this[0], %referenced_resource_type)
            <<Local Stack reference was invalid>>
        } or Ref {
            query_for_resource(%doc, this, %referenced_resource_type)
            <<Local Stack reference was invalid>>
        }
    }
}

rule query_for_resource(doc, resource_key, referenced_resource_type) {

CT.CODEBUILD.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
let referenced_resource = %doc.Resources[ keys == %resource_key ]
%referenced_resource not empty
%referenced_resource {
    Type == %referenced_resource_type
}
```
CloudWatchLogs:
   Status: ENABLED

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
CodeBuildServiceRole:
   Type: AWS::IAM::Role
   Properties:
      AssumeRolePolicyDocument:
         Version: '2012-10-17'
         Statement:
            - Effect: Allow
              Principal:
                 Service: codebuild.amazonaws.com
              Action: sts:AssumeRole
CodeBuildProject:
   Type: AWS::CodeBuild::Project
   Properties:
      Artifacts:
         Type: NO_ARTIFACTS
      Environment:
         ComputeType: BUILD_GENERAL1_SMALL
         Image: aws/codebuild/standard:4.0
         Type: LINUX_CONTAINER
      ServiceRole:
         Fn::GetAtt:
            - CodeBuildServiceRole
            - Arn
      Source:
         Type: NO_SOURCE
      BuildSpec: |
         version: 0.2
         phases:
            install:
               commands:
                  - npm install
            build:
               commands:
                  - npm test
      LogsConfig:
         S3Logs:
            Status: DISABLED
         CloudWatchLogs:
            Status: DISABLED

[CT.CODEBUILD.PR.4] Require any AWS CodeBuild project to deactivate privileged mode when running

This control checks whether AWS CodeBuild projects have privileged mode turned off.

- **Control objective:** Enforce least privilege
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CodeBuild::Project
- **AWS CloudFormation guard rule:** CT.CODEBUILD.PR.4 rule specification (p. 415)
Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.CODEBUILD.PR.4 rule specification (p. 415)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.CODEBUILD.PR.4 example templates (p. 417)

Explanation

By default, Docker containers do not allow access to any devices. Privileged mode grants a build project Docker container access to all devices. Setting privilegedMode with value true enables the Docker daemon to run inside a Docker container. The Docker daemon listens for Docker API requests, and it manages Docker objects such as images, containers, networks, and volumes. This parameter should be set to true if the build project is intended to build Docker images. Otherwise, this setting should be deactivated, to prevent unintended access to Docker APIs, or to the container's underlying hardware. Unintended access to privilegedMode may expose your system to risk of malicious tampering, or deletion of critical resources.

Remediation for rule failure

Within Environment, set PrivilegedMode to false or omit the PrivilegedMode property.

The examples that follow show how to implement this remediation.

AWS CodeBuild Project - Example One

AWS CodeBuild project configured to deactivate privileged mode, by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.

JSON example

```json
{
   "CodeBuildProject": {
      "Type": "AWS::CodeBuild::Project",
      "Properties": {
         "Environment": {
            "ComputeType": "BUILD_GENERAL1_SMALL",
            "Image": "aws/codebuild/standard:4.0",
            "Type": "LINUX_CONTAINER"
         },
         "ServiceRole": {
            "Fn::GetAtt": [
               "CodeBuildServiceRole",
               "Arn"
            ]
         },
         "Source": {
            "Type": "NO_SOURCE",
            "BuildSpec": "version: 0.2
install:
  commands:
  - npm install
build:
  commands:
  - npm test
artifacts:
  files:
  - **/*
   }
   }

YAML example

CodeBuildProject:
```
The examples that follow show how to implement this remediation.

**AWS CodeBuild Project - Example Two**

AWS CodeBuild project configured to deactivate privileged mode, by means of the `PrivilegedMode` property. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "CodeBuildProject": {
    "Type": "AWS::CodeBuild::Project",
    "Properties": {
      "Environment": {
        "ComputeType": "BUILD_GENERAL1_SMALL",
        "Image": "aws/codebuild/standard:4.0",
        "Type": "LINUX_CONTAINER",
        "PrivilegedMode": false
      },
      "ServiceRole": {
        "Fn::GetAtt": [
          "CodeBuildServiceRole",
          "Arn"
        ]
      },
      "Source": {
        "Type": "NO_SOURCE",
        "BuildSpec": "version: 0.2
                  phases:
                  install:
                    commands: 
                      - npm install
                  build:
                    commands:
                      - npm test
                  artifacts:
                    files: 
                      - '***/**'
        
```
CT.CODEBUILD.PR.4 rule specification

# Province: AWS::CodeBuild::Project
Type: AWS::CodeBuild::Project
Properties:

Environment:
  ComputeType: BUILD_GENERAL1_SMALL
  Image: aws/codebuild/standard:4.0
  Type: LINUX_CONTAINER
  PrivilegedMode: false
  ServiceRole: !GetAtt 'CodeBuildServiceRole.Arn'

Source:
  Type: NO_SOURCE
  BuildSpec: |
      version: 0.2
      phases:
        install:
          commands:
            - npm install
        build:
          commands:
            - npm test
      artifacts:
        files:
            - '**/*'

# Province: AWS::CodeBuild::Project
Type: AWS::CodeBuild::Project
Properties:

Environment:
  ComputeType: BUILD_GENERAL1_SMALL
  Image: aws/codebuild/standard:4.0
  Type: LINUX_CONTAINER
  PrivilegedMode: false
  ServiceRole: !GetAtt 'CodeBuildServiceRole.Arn'

Source:
  Type: NO_SOURCE
  BuildSpec: |
      version: 0.2
      phases:
        install:
          commands:
            - npm install
        build:
          commands:
            - npm test
      artifacts:
        files:
            - '**/*'
# And: 'Environment' configuration is provided
# And: 'PrivilegedMode' within the 'Environment' configuration is provided and set to bool(true)
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains a CodeBuild project resource
#   And: 'Environment' configuration is provided
#   And: 'PrivilegedMode' within 'Environment' configuration is not provided
# Then: PASS
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains a CodeBuild project resource
#   And: 'Environment' configuration is provided
#   And: 'PrivilegedMode' within 'Environment' configuration is set to bool(false)
# Then: PASS

# Constants

let CODEBUILD_PROJECT_TYPE = "AWS::CodeBuild::Project"
let INPUT_DOCUMENT = this

# Assignments

let codebuild_projects = Resources.*[ Type == %CODEBUILD_PROJECT_TYPE ]

# Primary Rules

rule codebuild_project_environment_privileged_check when is_cfn_template(%INPUT_DOCUMENT) {
  %codebuild_projects not empty {
    check(%codebuild_projects.Properties)
    <<
      [CT.CODEBUILD.PR.4]: Require any AWS CodeBuild project to deactivate privileged mode when running
      [FIX]: Within 'Environment', set 'PrivilegedMode' to 'false' or omit the 'PrivilegedMode' property.
    >>
  }
}

rule codebuild_project_environment_privileged_check when is_cfn_hook(%INPUT_DOCUMENT, %CODEBUILD_PROJECT_TYPE) {
  check(%INPUT_DOCUMENT.%CODEBUILD_PROJECT_TYPE.resourceProperties)
  <<
    [CT.CODEBUILD.PR.4]: Require any AWS CodeBuild project to deactivate privileged mode when running
    [FIX]: Within 'Environment', set 'PrivilegedMode' to 'false' or omit the 'PrivilegedMode' property.
  >>
}

# Parameterized Rules

rule check(codebuild_project) {
  %codebuild_project {
    # Scenario 2
    Environment exists
    Environment is_struct
    Environment {
      # Scenario 4
      PrivilegedMode not exists or
# Scenario 3 and 5
PrivilegedMode == false

# Utility Rules

CT.CODEBUILD.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  CodeBuildServiceRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: codebuild.amazonaws.com
            Action: sts:AssumeRole
  CodeBuildProject:
    Type: AWS::CodeBuild::Project
    Properties:
      Artifacts:
        Type: NO_ARTIFACTS
      Environment:
        ComputeType: BUILD_GENERAL1_SMALL
        Image: aws/codebuild/standard:4.0
        Type: LINUX_CONTAINER
      ServiceRole:
        Fn::GetAtt:
        - CodeBuildServiceRole
        - Arn
      Source:
        Type: NO_SOURCE
        BuildSpec: |
          version: 0.2
          phases:
            install:
              commands: [npm install]
            build:
              commands: [npm test]
        artifacts:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

<table>
<thead>
<tr>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>CodeBuildServiceRole:</td>
</tr>
<tr>
<td>Type: AWS::IAM::Role</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>AssumeRolePolicyDocument:</td>
</tr>
<tr>
<td>Version: '2012-10-17'</td>
</tr>
<tr>
<td>Statement:</td>
</tr>
<tr>
<td>- Effect: Allow</td>
</tr>
<tr>
<td>Principal:</td>
</tr>
<tr>
<td>Service: codebuild.amazonaws.com</td>
</tr>
<tr>
<td>Action: sts:AssumeRole</td>
</tr>
<tr>
<td>CodeBuildProject:</td>
</tr>
<tr>
<td>Type: AWS::CodeBuild::Project</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>Artifacts:</td>
</tr>
<tr>
<td>Type: NO_ARTIFACTS</td>
</tr>
<tr>
<td>Environment:</td>
</tr>
<tr>
<td>ComputeType: BUILD_GENERAL1_SMALL</td>
</tr>
<tr>
<td>Image: aws/codebuild/standard:4.0</td>
</tr>
<tr>
<td>Type: LINUX_CONTAINER</td>
</tr>
<tr>
<td>PrivilegedMode: true</td>
</tr>
<tr>
<td>ServiceRole:</td>
</tr>
<tr>
<td>Fn::GetAtt:</td>
</tr>
<tr>
<td>- CodeBuildServiceRole</td>
</tr>
<tr>
<td>- Arn</td>
</tr>
<tr>
<td>Source:</td>
</tr>
<tr>
<td>Type: NO_SOURCE</td>
</tr>
<tr>
<td>BuildSpec:</td>
</tr>
<tr>
<td>version: 0.2</td>
</tr>
<tr>
<td>phases:</td>
</tr>
<tr>
<td>install:</td>
</tr>
<tr>
<td>commands:</td>
</tr>
<tr>
<td>- npm install</td>
</tr>
<tr>
<td>build:</td>
</tr>
<tr>
<td>commands:</td>
</tr>
<tr>
<td>- npm test</td>
</tr>
<tr>
<td>artifacts:</td>
</tr>
<tr>
<td>files:</td>
</tr>
<tr>
<td>- '**<em>/</em>'</td>
</tr>
</tbody>
</table>

**[CT.CODEBUILD.PR.5] Require encryption on all AWS CodeBuild project artifacts**

This control checks whether AWS CodeBuild projects are configured to encrypt artifacts.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CodeBuild::Project
- **AWS CloudFormation guard rule:** [CT.CODEBUILD.PR.5 rule specification](p. 422)

**Details and examples**
Proactive controls

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CODEBUILD.PR.5 rule specification (p. 422)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CODEBUILD.PR.5 example templates (p. 424)]

Explanation

Encryption of data at rest is a recommended best practice. It adds a layer of access management around your data. In case of a compromise to your CodeBuild artifacts, encryption at rest ensures that your data is protected from unintended access.

Usage considerations

- This control applies only to AWS CodeBuild projects configured to return primary or secondary artifacts as output.

Remediation for rule failure

Set the EncryptionDisabled property in Artifacts and any SecondaryArtifacts to false, or omit the EncryptionDisabled property.

The examples that follow show how to implement this remediation.

AWS CodeBuild Project - Example One

AWS CodeBuild project configured to return primary artifacts as output with artifact encryption enabled, by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.

JSON example

```json
{
  "CodeBuildProject": {
    "Type": "AWS::CodeBuild::Project",
    "Properties": {
      "Environment": {
        "ComputeType": "BUILD_GENERAL1_SMALL",
        "Image": "aws/codebuild/standard:4.0",
        "Type": "LINUX_CONTAINER"
      },
      "ServiceRole": {
        "Fn::GetAtt": [
          "CodeBuildServiceRole",
          "Arn"
        ]
      },
      "Source": {
        "Type": "NO_SOURCE",
        "BuildSpec": "version: 0.2\ninstall\nbuild\ncommands:\n-npm install\n-npm test\nants:\nfiles:\n-**/*\n",
        "Artifacts": {
          "Type": "S3",
          "Location": {
            "Ref": "S3Bucket"
          }
        }
      }
    }
  }
}
```
YAML example

CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Environment:
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:4.0
      Type: LINUX_CONTAINER
    ServiceRole: !GetAtt 'CodeBuildServiceRole.Arn'
  Source:
    Type: NO_SOURCE
    BuildSpec: |
      version: 0.2
      phases:
        install:
          commands:
            - npm install
        build:
          commands:
            - npm test
        artifacts:
          files:
            - '**/*'
      Artifacts:
        Type: S3
        Location: !Ref 'S3Bucket'

The examples that follow show how to implement this remediation.

AWS CodeBuild Project - Example Two

AWS CodeBuild project configured to return primary and secondary artifacts as output with artifact encryption enabled, by means of the EncryptionDisabled property. The example is shown in JSON and in YAML.

JSON example

```json
{
  "CodeBuildProject": {
    "Type": "AWS::CodeBuild::Project",
    "Properties": {
      "Environment": {
        "ComputeType": "BUILD_GENERAL1_SMALL",
        "Image": "aws/codebuild/standard:4.0",
        "Type": "LINUX_CONTAINER"
      },
      "ServiceRole": {
        "Fn::GetAtt": [
          "CodeBuildServiceRole",
          "Arn"
        ]
      },
      "Source": {
        "Type": "NO_SOURCE",
        "BuildSpec": "version: 0.2\ninstall:\n  commands:\n    - npm install\nbuild:\n  commands:\n    - npm test\nartifacts:\n  files:\n    - '**/*'
secondary-artifacts:\n  secondaryArtifact:\n    files:\n      - 'directory/file1'
```
"Artifacts": {  
    "Type": "S3",  
    "EncryptionDisabled": false,  
    "Location": {  
        "Ref": "S3Bucket"  
    }  
},  
"SecondaryArtifacts": [  
    {  
        "Type": "S3",  
        "EncryptionDisabled": false,  
        "ArtifactIdentifier": "secondaryArtifact",  
        "Location": {  
            "Ref": "S3Bucket"  
        }  
    }  
]  
}  

YAML example

CodeBuildProject:  
  Type: AWS::CodeBuild::Project  
  Properties:  
    Environment:  
      ComputeType: BUILD_GENERAL1_SMALL  
      Image: aws/codebuild/standard:4.0  
      Type: LINUX_CONTAINER  
      ServiceRole: !GetAtt 'CodeBuildServiceRole.Arn'  
    Source:  
      Type: NO_SOURCE  
      BuildSpec: |  
        version: 0.2  
        phases:  
          install:  
            commands:  
              - npm install  
          build:  
            commands:  
              - npm test  
          artifacts:  
            files:  
              - '***/*'  
            secondary-artifacts:  
              secondaryArtifact:  
                files:  
                  - 'directory/file1'  
        Artsifacts:  
          Type: S3  
          EncryptionDisabled: false  
          Location: !Ref 'S3Bucket'  
          SecondaryArtifacts:  
            - Type: S3  
              EncryptionDisabled: false  
              ArtifactIdentifier: secondaryArtifact  
              Location: !Ref 'S3Bucket'
CT.CODEBUILD.PR.5 rule specification

# ##############################################################################
##       Rule Specification        ##
# ##############################################################################

# Rule Identifier:
#   codebuild_project_artifact_encryption_check
# Description:
#   This control checks whether AWS CodeBuild projects are configured to encrypt artifacts.
# Reports on:
#   AWS::CodeBuild::Project
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
#   None
# Scenarios:
#   Scenario 1:
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any CodeBuild project resources
#     Then: SKIP
#   Scenario 2:
#     Given: The input document contains a CodeBuild project resource
#     And: 'Artifacts' configuration is provided and is of 'Type' 'NO_ARTIFACTS'
#     And: 'SecondaryArtifacts' configuration is not provided or provided with an empty list
#     Then: SKIP
#   Scenario 3:
#     Given: The input document contains a CodeBuild project resource
#     And: 'Artifacts' configuration is provided and is of 'Type' 'NO_ARTIFACTS'
#     And: 'SecondaryArtifacts' configuration is provided as a non-empty list
#     And: All 'SecondaryArtifacts' entries have 'Type' set to 'NO_ARTIFACTS'
#     Then: SKIP
#   Scenario 4:
#     Given: The input document contains a CodeBuild project resource
#     And: 'Artifacts' configuration is provided and is not of 'Type' 'NO_ARTIFACTS'
#     And: 'EncryptionDisabled' within 'Artifacts' configuration is provided and set to bool(true)
#     Then: FAIL
#   Scenario 5:
#     Given: The input document contains a CodeBuild project resource
#     And: 'SecondaryArtifacts' configuration is provided
#     And: There exists one or more items in 'SecondaryArtifacts' which have 'EncryptionDisabled' set to bool(true)
#     Then: FAIL
#   Scenario 6:
#     Given: The input document contains a CodeBuild project resource
#     And: 'Artifacts.EncryptionDisabled' is not provided, or is set to bool(false)
And: There exists no item in 'SecondaryArtifacts' which has 'EncryptionDisabled' set to bool(true)
Then: PASS

# Constants
#
let CODEBUILD_PROJECT_TYPE = "AWS::CodeBuild::Project"
let INPUT_DOCUMENT = this
#
# Assignments
#
let codebuild_project = Resources.*[ Type == %CODEBUILD_PROJECT_TYPE ]
#
# Primary Rules
#
rule codebuild_project_artifact_encryption_check when is_cfn_template(%INPUT_DOCUMENT)
%codebuild_project not empty {
  check(%codebuild_project.Properties)
<<
  [CT.CODEBUILD.PR.5]: Require encryption on all AWS CodeBuild project artifacts
  [FIX]: Set the 'EncryptionDisabled' property in 'Artifacts' and any 'SecondaryArtifacts' to 'false', or omit the 'EncryptionDisabled' property.
>>
}
rule codebuild_project_artifact_encryption_check when is_cfn_hook(%INPUT_DOCUMENT, %CODEBUILD_PROJECT_TYPE) {
  check(%INPUT_DOCUMENT.%CODEBUILD_PROJECT_TYPE.resourceProperties)
<<
  [CT.CODEBUILD.PR.5]: Require encryption on all AWS CodeBuild project artifacts
  [FIX]: Set the 'EncryptionDisabled' property in 'Artifacts' and any 'SecondaryArtifacts' to 'false', or omit the 'EncryptionDisabled' property.
>>
}
#
# Parameterized Rules
#
rule check(codebuild_project) {
  %codebuild_project [ filter_codebuild_projects(this) ] {
    Artifacts {
      # Scenario 4 and 6
      check_artifact(this)
    }
    # Scenario 5
    SecondaryArtifacts not exists or
    check_secondary_artifacts(this)
  }
}
rule check_secondary_artifacts(codebuild_project) {
  %codebuild_project {
    SecondaryArtifacts is_list
    SecondaryArtifacts[*] {
      # Scenario 5 and 6
      check_artifact(this)
    }
  }
}
rule check_artifact(artifact) {

%artifact {
    EncryptionDisabled not exists or
    EncryptionDisabled == false
}
}

rule filter_codebuild_projects(codebuild_project) {
    %codebuild_project {
        # Scenario 2 and 3
        Artifacts exists
        Artifacts is_struct
        Artifacts {
            filter_artifact(this)
        } or
        filter_secondary_artifacts(this)
    }
}

rule filter_secondary_artifacts(codebuild_project) {
    %codebuild_project {
        # Scenario 2
        SecondaryArtifacts exists
        SecondaryArtifacts is_list
        SecondaryArtifacts not empty
        SecondaryArtifacts[*] {
        # Scenario 3
            filter_artifact(this)
        }
    }
}

rule filter_artifact(artifact) {
    %artifact {
        Type exists
        Type != "NO_ARTIFACTS"
    }
}

# Utility Rules
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.CODEBUILD.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
CodeBuildServiceRole:
    Type: AWS::IAM::Role
Properties:
AssumeRolePolicyDocument:
  Version: '2012-10-17'
  Statement:
    - Effect: Allow
    Principal:
      Service: codebuild.amazonaws.com
    Action: sts:AssumeRole
  Path: /
Policies:
  - PolicyName: CodeBuildProjectPolicy
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
        Action:
          - logs:CreateLogGroup
          - logs:CreateLogStream
          - logs:PutLogEvents
        Resource: '*'
        - Effect: Allow
        Action:
          - s3:PutObject
          - s3:GetBucketAcl
          - s3:GetBucketLocation
        Resource:
          - Fn::GetAtt:
            - S3Bucket
            - Arn
          - Fn::Join:
            - ''
            - [Fn::GetAtt:
              - S3Bucket
              - Arn
              - '/*']
S3Bucket:
  Type: AWS::S3::Bucket
  Properties: {}
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Environment:
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:4.0
      Type: LINUX_CONTAINER
    ServiceRole:
      Fn::GetAtt:
        - CodeBuildServiceRole
        - Arn
    Source:
      Type: NO_SOURCE
      BuildSpec:
        version: 0.2
        phases:
          install:
            commands:
              - npm install
          build:
            commands:
              - npm test
        artifacts:
          files:
            - '***/*'
Artifacts:
  Type: S3
  Location:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
CodeBuildServiceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal: codebuild.amazonaws.com
          Action: sts:AssumeRole
          Path: /
        Policies:
          - PolicyName: CodeBuildProjectPolicy
            PolicyDocument:
              Version: '2012-10-17'
              Statement:
                - Effect: Allow
                  Action:
                    - logs:CreateLogGroup
                    - logs:CreateLogStream
                    - logs:PutLogEvents
                  Resource: '*'
                - Effect: Allow
                  Action:
                    - s3:PutObject
                    - s3:GetBucketAcl
                    - s3:GetBucketLocation
                  Resource:
                    - Fn::GetAtt:
                      - S3Bucket
                      - Arn
                    - Fn::Join:
                      - ''
                      - - Fn::GetAtt:
                          - S3Bucket
                          - Arn
                      - ''
                      - '/**'

S3Bucket:
  Type: AWS::S3::Bucket
  Properties: {}

CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Environment:
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:4.0
      Type: LINUX_CONTAINER
    ServiceRole:
      Fn::GetAtt:
        - CodeBuildServiceRole
        - Arn
    Source:
      Type: NO_SOURCE
      BuildSpec:
        version: 0.2
        phases:
Proactive controls

install:
  commands:
    - npm install
build:
  commands:
    - npm test
artifacts:
  files:
    - '**/*'
Artifacts:
  Type: S3
  EncryptionDisabled: true
  Location:
    Ref: S3Bucket

[CT.CODEBUILD.PR.6] Require encryption on all Amazon S3 logs for AWS CodeBuild projects

This control checks whether AWS CodeBuild projects configured with Amazon S3 logs have encryption enabled.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::CodeBuild::Project
- **AWS CloudFormation guard rule:** [CT.CODEBUILD.PR.6 rule specification (p. 430)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.CODEBUILD.PR.6 rule specification (p. 430)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.CODEBUILD.PR.6 example templates (p. 432)]

Explanation

Encryption of data at rest is a recommended best practice. It adds a layer of access management around your data. In case of a compromise to your CodeBuild artifacts, encryption at rest ensures that your data is protected from unintended access.

**Usage considerations**

- This control applies only to AWS CodeBuild projects with log delivery to Amazon S3 enabled.

Remediation for rule failure

Set EncryptionDisabled in S3Logs to false, or do not specify the EncryptionDisabled property.

The examples that follow show how to implement this remediation.

**AWS CodeBuild Project - Example One**

AWS CodeBuild project configured to encrypt logs delivered to an Amazon S3 logging destination, by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.
**JSON example**

```json
{
    "CodeBuildProject": {
        "Type": "AWS::CodeBuild::Project",
        "Properties": {
            "Artifacts": {
                "Type": "NO_ARTIFACTS"
            },
            "Environment": {
                "ComputeType": "BUILD_GENERAL1_SMALL",
                "Image": "aws/codebuild/standard:4.0",
                "Type": "LINUX_CONTAINER"
            },
            "ServiceRole": {
                "Fn::GetAtt": [
                    "CodeBuildServiceRole",
                    "Arn"
                ]
            },
            "Source": {
                "Type": "NO_SOURCE",
                "BuildSpec": "version: 0.2
phases:
  install:
    commands:
      - npm install
  build:
    commands:
      - npm test
artifacts:
  files:
    - '**/*'
            }
        }
    }
}
```

**YAML example**

```yaml
CodeBuildProject:
  Type: AWS::CodeBuild::Project
Properties:
  Artifacts:
    Type: NO_ARTIFACTS
Environment:
  ComputeType: BUILD_GENERAL1_SMALL
  Image: aws/codebuild/standard:4.0
  Type: LINUX_CONTAINER
ServiceRole: !GetAtt 'CodeBuildServiceRole.Arn'
Source:
  Type: NO_SOURCE
BuildSpec: |
  version: 0.2
  phases:
    install:
      commands:
        - npm install
    build:
      commands:
        - npm test
  artifacts:
```
files:
   - '**/*'
LogsConfig:
S3Logs:
   Status: ENABLED
   Location: !Ref 'S3Bucket'

The examples that follow show how to implement this remediation.

**AWS CodeBuild Project - Example Two**

AWS CodeBuild project configured to encrypt logs delivered to an Amazon S3 logging destination, by means of the EncryptionDisabled property. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "CodeBuildProject": {
      "Type": "AWS::CodeBuild::Project",
      "Properties": {
         "Artifacts": {
            "Type": "NO_ARTIFACTS"
         },
         "Environment": {
            "ComputeType": "BUILD_GENERAL1_SMALL",
            "Image": "aws/codebuild/standard:4.0",
            "Type": "LINUX_CONTAINER"
         },
         "ServiceRole": {
            "Fn::GetAtt": [
               "CodeBuildServiceRole",
               "Arn"
            ]
         },
         "Source": {
            "Type": "NO_SOURCE",
            "BuildSpec": "version: 0.2
phases:
  install:
    commands:
      - npm install
  build:
    commands:
      - npm test
artifacts:
  files:
    - '**/*'
         }
      },
      "LogsConfig": {
         "S3Logs": {
            "Status": "ENABLED",
            "Location": {
               "Ref": "S3Bucket"
            },
            "EncryptionDisabled": false
         }
      }
   }
}
```

**YAML example**

```yaml
CodeBuildProject:
   Type: AWS::CodeBuild::Project
   Properties:
      Artifacts:
```
CT.CODEBUILD.PR.6 rule specification

```
# ################################################################
##   Rule Specification   ##
# ################################################################
#
# Rule Identifier:
#   codebuild_project_s3_logs_encrypted_check
#
# Description:
#   This control checks whether AWS CodeBuild projects configured with Amazon S3 logs have encryption enabled.
#
# Reports on:
#   AWS::CodeBuild::Project
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario 1:
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any CodeBuild project resources
#     Then: SKIP
#   Scenario 2:
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a CodeBuild project resource
#     And: 'S3Logs' in 'LogsConfig' configuration is not provided
#     Then: SKIP
#   Scenario 3:
```
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a CodeBuild project resource
# And: 'S3Logs' in 'LogsConfig' configuration is provided and its 'Status' is set to
'DISABLED'
# Then: SKIP

# Scenario 4:
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a CodeBuild project resource
# And: 'S3Logs' in 'LogsConfig' configuration is provided and its 'Status' is set to
'ENABLED'
# And: 'EncryptionDisabled' within 'S3Logs' is provided and set to bool(true)
# Then: FAIL

# Scenario 5:
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a CodeBuild project resource
# And: 'S3Logs' in 'LogsConfig' configuration is provided and its 'Status' is set to
'ENABLED'
# And: 'EncryptionDisabled' within 'S3Logs' is not provided
# Then: PASS

# Scenario 6:
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a CodeBuild project resource
# And: 'S3Logs' in 'LogsConfig' configuration is provided and its 'Status' is set to
'ENABLED'
# And: 'EncryptionDisabled' within 'S3Logs' is provided and set to bool(false)
# Then: PASS

# Constants

let CODEBUILD_PROJECT_TYPE = "AWS::CodeBuild::Project"
let INPUT_DOCUMENT = this

# Assignments

let codebuild_project = Resources.*[ Type == %CODEBUILD_PROJECT_TYPE ]

# Primary Rules

# rule codebuild_project_s3_logs_encrypted_check when is_cfn_template(%INPUT_DOCUMENT)
# %codebuild_project not empty {
#   check(%codebuild_project.Properties)
#     <<
#       [CT.CODEBUILD.PR.6]: Require encryption on all Amazon S3 logs for AWS CodeBuild
#       projects
#       [FIX]: Set 'EncryptionDisabled' in 'S3Logs' to 'false', or do not specify the
#       'EncryptionDisabled' property.
#     >>
# }

rule codebuild_project_s3_logs_encrypted_check when is_cfn_hook(%INPUT_DOCUMENT,
%CODEBUILD_PROJECT_TYPE) {
  check(%INPUT_DOCUMENT.%CODEBUILD_PROJECT_TYPE.resourceProperties)
  <<
    [CT.CODEBUILD.PR.6]: Require encryption on all Amazon S3 logs for AWS CodeBuild
    projects
    [FIX]: Set 'EncryptionDisabled' in 'S3Logs' to 'false', or do not specify the
    'EncryptionDisabled' property.
  >>
}

431
# Parameterized Rules

```python
# Scenario 2 and 3
filter_codebuild_projects(this) {
    LogsConfig {
        S3Logs {
            EncryptionDisabled not exists or
            EncryptionDisabled == false
        }
    }
}
```

```python
rule filter_codebuild_projects(codebuild_project) {
    %codebuild_project {
        LogsConfig exists
        LogsConfig is_struct
        LogsConfig {
            S3Logs exists
            S3Logs is_struct
            S3Logs {
                Status exists
                Status == "ENABLED"
            }
        }
    }
}
```

# Utility Rules

```python
# Scenario 2 and 3
filter_codebuild_projects(this) {
    LogsConfig {
        S3Logs {
            EncryptionDisabled not exists or
            EncryptionDisabled == false
        }
    }
}
```

```python
# Scenario 3 and 4
filter_codebuild_projects(this) {
    LogsConfig {
        S3Logs {
            Status exists
            Status == "ENABLED"
        }
    }
}
```

```python
# Utility Rules

```python
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
```

```python
rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

CT.CODEBUILD.PR.6 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  CodeBuildServiceRole:
    Type: AWS::IAM::Role
    Properties:
```
AssumeRolePolicyDocument:
  Version: '2012-10-17'
  Statement:
    - Effect: Allow
      Principal:
        Service: codebuild.amazonaws.com
      Action: sts:AssumeRole
      Path: /
      Policies:
        - PolicyName: CodeBuildProjectPolicy
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Effect: Allow
                Action:
                  - logs:CreateLogGroup
                  - logs:CreateLogStream
                  - logs:PutLogEvents
                Resource: '*'
              - Effect: Allow
                Action:
                  - s3:PutObject
                  - s3:GetBucketAcl
                  - s3:GetBucketLocation
                Resource:
                  - Fn::GetAtt:
                    - S3Bucket
                    - Arn
                  - Fn::Join:
                    - ''
                    - - Fn::GetAtt:
                      - S3Bucket
                      - Arn
                    - '/'
S3Bucket:
  Type: AWS::S3::Bucket
  Properties: {}
CodeBuildProject:
  Type: AWS::CodeBuild::Project
  Properties:
    Artifacts:
      Type: NO_ARTIFACTS
    Environment:
      ComputeType: BUILD_GENERAL1_SMALL
      Image: aws/codebuild/standard:4.0
      Type: LINUX_CONTAINER
    ServiceRole:
      Fn::GetAtt:
        - CodeBuildServiceRole
        - Arn
    Source:
      Type: NO_SOURCE
    BuildSpec:
      version: 0.2
      phases:
        install:
          commands:
            - npm install
        build:
          commands:
            - npm test
          artifacts:
            files:
              - '**/*'
    LogsConfig:
      S3Logs:
Status: ENABLED
Location: Ref: S3Bucket

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  CodeBuildServiceRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: codebuild.amazonaws.com
            Action: sts:AssumeRole
            Path: /
        Policies:
          - PolicyName: CodeBuildProjectPolicy
            PolicyDocument:
              Version: '2012-10-17'
              Statement:
                - Effect: Allow
                  Action:
                    - logs:CreateLogGroup
                    - logs:CreateLogStream
                    - logs:PutLogEvents
                  Resource: '*'
                - Effect: Allow
                  Action:
                    - s3:PutObject
                    - s3:GetBucketAcl
                    - s3:GetBucketLocation
                  Resource:
                    - Fn::GetAtt:
                      - S3Bucket
                      - Arn
                    - Fn::Join:
                      - ''
                      - - Fn::GetAtt:
                        - S3Bucket
                        - Arn
                      - '/*'
  S3Bucket:
    Type: AWS::S3::Bucket
    Properties: {}
  CodeBuildProject:
    Type: AWS::CodeBuild::Project
    Properties:
      Artifacts:
        Type: NO_ARTIFACTS
      Environment:
        ComputeType: BUILD_GENERAL1_SMALL
        Image: aws/codebuild/standard:4.0
        Type: LINUX_CONTAINER
      ServiceRole:
        Fn::GetAtt:
          - CodeBuildServiceRole
          - Arn
      Source:
        Type: NO_SOURCE
AWS Database Migration Service (AWS DMS) controls

Topics

• [CT.DMS.PR.1] Require that a public AWS DMS replication instance is not public (p. 435)

[CT.DMS.PR.1] Require that a public AWS DMS replication instance is not public

This control checks whether your AWS DMS replication instance is public.

• Control objective: Limit network access, Enforce least privilege
• Implementation: AWS CloudFormation Guard Rule
• Control behavior: Proactive
• Resource types: AWS::DMS::ReplicationInstance
• AWS CloudFormation guard rule: CT.DMS.PR.1 rule specification (p. 436)

Details and examples

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.DMS.PR.1 rule specification (p. 436)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.DMS.PR.1 example templates (p. 438)

Explanation

A private replication instance has a private IP address that you cannot access outside of the replication network. You use a private instance when both source and target databases are in the same network that is connected to the replication instance's VPC. The network can be connected to the VPC by using a VPN, AWS Direct Connect, or VPC peering.

Remediation for rule failure

Set PubliclyAccessible to false.

The examples that follow show how to implement this remediation.
**AWS DMS Replication Instance - Example**

AWS DMS replication instance configured with public access disabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "DMSReplicationInstance": {
      "Type": "AWS::DMS::ReplicationInstance",
      "Properties": {
         "ReplicationInstanceClass": "dms.t3.micro",
         "PubliclyAccessible": false
      }
   }
}
```

**YAML example**

```yaml
DMSReplicationInstance:
   Type: AWS::DMS::ReplicationInstance
   Properties:
      ReplicationInstanceClass: dms.t3.micro
      PubliclyAccessible: false
```

**CT.DMS.PR.1 rule specification**

```plaintext
# ###################################
##       Rule Specification        ##
#####################################
# Rule Identifier:
#   dms_replication_instance_not_public_check
#
# Description:
#   This control checks whether your AWS DMS replication instance is public.
#
# Reports on:
#   AWS::DMS::ReplicationInstance
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#      And: The input document does not contain any AWS DMS replication instance resources
#      Then: SKIP
#   Scenario: 2
#      Given: The input document contains a AWS DMS replication instance resource
#      And: 'PubliclyAccessible' is not present on the AWS DMS replication instance
```

436
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a AWS DMS replication instance resource
# And: 'PubliclyAccessible' is present on the AWS DMS replication instance
# And: 'PubliclyAccessible' is set to bool(true)
# Then: FAIL

# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a AWS DMS replication instance resource
# And: 'PubliclyAccessible' is present on the AWS DMS replication instance
# And: 'PubliclyAccessible' is set to bool(false)
# Then: PASS

# Constants

let DMS_REPLICATION_INSTANCE_TYPE = "AWS::DMS::ReplicationInstance"
let INPUT_DOCUMENT = this

# Assignments

let dms_replication_instances = Resources.*[ Type == %DMS_REPLICATION_INSTANCE_TYPE ]

# Primary Rules

rule dms_replication_instance_not_public_check when is_cfn_template(%INPUT_DOCUMENT) {
    %dms_replication_instances not empty {
        check(%dms_replication_instances.Properties)
        <<
            [CT.DMS.PR.1]: Require that a public AWS DMS replication instance is not public
            [FIX]: Set 'PubliclyAccessible' to 'false'.
        >>
    }
}

rule dms_replication_instance_not_public_check when is_cfn_hook(%INPUT_DOCUMENT, %DMS_REPLICATION_INSTANCE_TYPE) {
    check(%INPUT_DOCUMENT.%DMS_REPLICATION_INSTANCE_TYPE.resourceProperties)
    <<
        [CT.DMS.PR.1]: Require that a public AWS DMS replication instance is not public
        [FIX]: Set 'PubliclyAccessible' to 'false'.
    >>
}

# Parameterized Rules

rule check(dms_replication_instances) {
    %dms_replication_instances {
        # Scenario 2
        PubliclyAccessible exists
        # Scenario 3 and 4
        PubliclyAccessible == false
    }
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
CT.DMS.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
DMSReplicationInstance:
  Type: AWS::DMS::ReplicationInstance
  Properties:
    ReplicationInstanceClass: dms.t3.micro
    PubliclyAccessible: false

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
DMSReplicationInstance:
  Type: AWS::DMS::ReplicationInstance
  Properties:
    ReplicationInstanceClass: dms.t3.micro
    PubliclyAccessible: true

Amazon DocumentDB controls

Topics
- [CT.DOCUMENTDB.PR.1] Require an Amazon DocumentDB cluster to be encrypted at rest (p. 438)
- [CT.DOCUMENTDB.PR.2] Require an Amazon DocumentDB cluster to have a backup retention period greater than or equal to seven days (p. 442)

[CT.DOCUMENTDB.PR.1] Require an Amazon DocumentDB cluster to be encrypted at rest

This control checks whether storage encryption is enabled for an Amazon DocumentDB (with MongoDB compatibility) cluster.

- Control objective: Encrypt data at rest
- Implementation: AWS CloudFormation guard rule
- Control behavior: Proactive
- Resource types: AWS::DocDB::DBCluster
- AWS CloudFormation guard rule: CT.DOCUMENTDB.PR.1 rule specification (p. 440)
Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.DOCUMENTDB.PR.1 rule specification (p. 440)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.DOCUMENTDB.PR.1 example templates (p. 441)

Explanation

You encrypt data at rest in your Amazon DocumentDB cluster by specifying the storage encryption option when you create your cluster. Storage encryption is enabled cluster-wide, and it is applied to all instances, including the primary instance and any replicas. It also is applied to your cluster’s storage volume, data, indexes, logs, automated backups, and snapshots.

Clusters that you create using AWS CloudFormation have encryption at rest turned off by default. Therefore, you must explicitly enable encryption at rest using the StorageEncrypted property.

Remediation for rule failure

Set the value of the StorageEncrypted parameter to true.

The examples that follow show how to implement this remediation.

Amazon DocumentDB Cluster - Example

An Amazon DocumentDB cluster configured with storage encryption enabled. The example is shown in JSON and in YAML.

JSON example

```json
{
    "DocumentDBCluster": {
        "Type": "AWS::DocDB::DBCluster",
        "Properties": {
            "MasterUsername": {
                "Fn::Sub": "{{resolve:secretsmanager: \${DocumentDBClusterSecret}:::username}}"
            },
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager: \${DocumentDBClusterSecret}:::password}}"
            },
            "StorageEncrypted": true
        }
    }
}
```

YAML example

```yaml
DocumentDBCluster:
  Type: AWS::DocDB::DBCluster
  Properties:
    MasterUsername: !Sub '{{resolve:secretsmanager: \${DocumentDBClusterSecret}:::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager: \${DocumentDBClusterSecret}:::password}}'
    StorageEncrypted: true
```

439
CT.DOCUMENTDB.PR.1 rule specification

```plaintext
# # Rule Specification #
#
# Rule Identifier:
#   docdb_cluster_encrypted_check
#
# Description:
#   This control checks whether storage encryption is enabled for an Amazon DocumentDB
#   (with MongoDB compatibility) cluster.
#
# Reports on:
#   AWS::DocDB::DBCluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#           And: The input document does not contain any Document DB cluster resources
#           Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#           And: The input document contains a Document DB cluster resource
#           And: 'StorageEncrypted' has not been provided
#           Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#           And: The input document contains a Document DB cluster resource
#           And: 'StorageEncrypted' has been provided and set to a value other than bool(true)
#           Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#           And: The input document contains a Document DB cluster resource
#           And: 'StorageEncrypted' has been provided and set to bool(true)
#           Then: PASS
#
# Constants
#
let DOCUMENT_DB_CLUSTER_TYPE = "AWS::DocDB::DBCluster"
let INPUT_DOCUMENT = this

# Assignments
#
let document_db_clusters = Resources.*[ Type == %DOCUMENT_DB_CLUSTER_TYPE ]

# Primary Rules
#
rule docdb_cluster_encrypted_check when is_cfn_template(%INPUT_DOCUMENT)
    %document_db_clusters not empty {
        check(%document_db_clusters.Properties)
    }
```

440
Proactive controls

<<
[CT.DOCUMENTDB.PR.1]: Require an Amazon DocumentDB cluster to be encrypted at rest
[FIX]: Set the value of the 'StorageEncrypted' parameter to true.
>>

rule docdb_cluster_encrypted_check when is_cfn_hook(%INPUT_DOCUMENT, %DOCUMENT_DB_CLUSTER_TYPE) {
  check(%INPUT_DOCUMENT.%DOCUMENT_DB_CLUSTER_TYPE.resourceProperties)
  <<
  [CT.DOCUMENTDB.PR.1]: Require an Amazon DocumentDB cluster to be encrypted at rest
  [FIX]: Set the value of the 'StorageEncrypted' parameter to true.
  >>
}

# Parameterized Rules
#
rule check(document_db_cluster) {
  %document_db_cluster {
    # Scenario 2
    StorageEncrypted exists
    # Scenarios 3 and 4
    StorageEncrypted == true
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists  or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.DOCUMENTDB.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
DocumentDBClusterSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    GenerateSecretString:
      SecretStringTemplate: '{"username": "exampleuser"}'
      GenerateStringKey: password
      PasswordLength: 16
      ExcludeCharacters: "@/\"

DocumentDBCluster:
  Type: AWS::DocDB::DBCluster
  Properties:
    MasterUsername:
      Fn::Sub: '{resolve:secretsmanager:${DocumentDBClusterSecret}::username}'}
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
DocumentDBClusterSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    GenerateSecretString:
      SecretStringTemplate: '{"username": "exampleuser"}'
      GenerateStringKey: password
      PasswordLength: 16
      ExcludeCharacters: "@/\"
DocumentDBCluster:
  Type: AWS::DocDB::DBCluster
  Properties:
    MasterUsername:
      Fn::Sub: '{resolve:secretsmanager:${DocumentDBClusterSecret}::username}'
    MasterUserPassword:
      Fn::Sub: '{resolve:secretsmanager:${DocumentDBClusterSecret}::password}'
    StorageEncrypted: false

[CTDOCUMENTDBPR2] Require an Amazon DocumentDB cluster to have a backup retention period greater than or equal to seven days

This control checks whether an Amazon DocumentDB cluster retention period is set to seven or more days (>=7). The default retention period is one day.

- **Control objective**: Improve resiliency
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::DocDB::DBCluster
- **AWS CloudFormation guard rule**: [CTDOCUMENTDBPR2 rule specification](p. 443)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CTDOCUMENTDBPR2 rule specification](p. 443)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CTDOCUMENTDBPR2 example templates](p. 445)

Explanation

Amazon DocumentDB creates daily automatic snapshots of your cluster during your cluster's backup window. Amazon DocumentDB saves the automatic snapshots of your cluster according to the backup retention period that you specify, allowing you to restore to any point within the backup retention period. This daily snapshot strengthens the resilience of your systems, and it can help you recover quickly from a security incident.
Remediation for rule failure

Set the value of the BackupRetentionPeriod parameter to an integer value between 7 and 35 days (inclusive).

The examples that follow show how to implement this remediation.

Amazon DocumentDB Cluster - Example

An Amazon DocumentDB cluster configured with a backup retention period of seven (7) days. The example is shown in JSON and in YAML.

JSON example

```
{
  "DocumentDBCluster": {
    "Type": "AWS::DocDB::DBCluster",
    "Properties": {
      "MasterUsername": {
        "Fn::Sub": "{{resolve:secretsmanager:${DocumentDBClusterSecret}::username}}"
      },
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${DocumentDBClusterSecret}::password}}"
      },
      "BackupRetentionPeriod": 7
    }
  }
}
```

YAML example

```
DocumentDBCluster:
  Type: AWS::DocDB::DBCluster
  Properties:
    MasterUsername: !Sub '{{resolve:secretsmanager:${DocumentDBClusterSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DocumentDBClusterSecret}::password}}'
    BackupRetentionPeriod: 7
```

CT.DOCUMENTDB.PR.2 rule specification

```
# ####################################################################
##       Rule Specification       
# ####################################################################
#
# Rule Identifier:
#   docdb_cluster_backup_retention_check
#
# Description:
#   This control checks whether an Amazon DocumentDB cluster retention period is set to seven or more days (>=7).
#
# Reports on:
#   AWS::DocDB::DBCluster
#
```

443
# Evaluates:
#  AWS CloudFormation, AWS CloudFormation hook
#  None
#
# Scenarios:
#  Scenario: 1
#  Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#  And: The input document does not contain any Document DB cluster resources
#  Then: SKIP
#  Scenario: 2
#  Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#  And: The input document contains a Document DB cluster resource
#  And: 'BackupRetentionPeriod' has not been provided
#  Then: FAIL
#  Scenario: 3
#  Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#  And: The input document contains a Document DB cluster resource
#  And: 'BackupRetentionPeriod' has been provided and set to an integer value less than seven (<7)
#  Then: FAIL
#  Scenario: 4
#  Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#  And: The input document contains a Document DB cluster resource
#  And: 'BackupRetentionPeriod' has been provided and set to an integer value greater than or equal to seven (>=7)
#  Then: PASS
#
# Constants
#
# let DOCUMENT_DB_CLUSTER_TYPE = "AWS::DocDB::DBCluster"
# let INPUT_DOCUMENT = this
#
# Assignments
#
# let document_db_clusters = Resources.*[ Type == %DOCUMENT_DB_CLUSTER_TYPE ]
#
# Primary Rules
#
rule docdb_cluster_backup_retention_check when is_cfn_template(%INPUT_DOCUMENT)
# %document_db_clusters not empty {
#   check(%document_db_clusters.Properties)
#   <<
#   [CT.DOCUMENTDB.PR.2]: Require an Amazon DocumentDB cluster to have automatic backups enabled
#   [FIX]: Set the value of the 'BackupRetentionPeriod' parameter to an integer value between 7 and 35 days (inclusive).
#   >>
# }

rule docdb_cluster_backup_retention_check when is_cfn_hook(%INPUT_DOCUMENT,%DOCUMENT_DB_CLUSTER_TYPE) {
  check(%INPUT_DOCUMENT.%DOCUMENT_DB_CLUSTER_TYPE.resourceProperties)
  <<
  [CT.DOCUMENTDB.PR.2]: Require an Amazon DocumentDB cluster to have automatic backups enabled
  [FIX]: Set the value of the 'BackupRetentionPeriod' parameter to an integer value between 7 and 35 days (inclusive).
CT.DOCUMENTDB.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  DocumentDBClusterSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      GenerateSecretString:
        SecretStringTemplate: '{"username": "exampleuser"}'
        GenerateStringKey: password
        PasswordLength: 16
        ExcludeCharacters: '\"@/\\'
  DocumentDBCluster:
    Type: AWS::DocDB::DBCluster
    Properties:
      MasterUsername:
        Fn::Sub: '{resolve:secretsmanager:{$DocumentDBClusterSecret::username}}'
      MasterUserPassword:
        Fn::Sub: '{resolve:secretsmanager:{$DocumentDBClusterSecret::password}}'
      BackupRetentionPeriod: 7

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  DocumentDBClusterSecret:
Amazon DynamoDB controls

Topics

- [CT.DYNAMODB.PR.1] Require that point-in-time recovery for an Amazon DynamoDB table is activated (p. 446)
- [CT.DYNAMODB.PR.2] Require an Amazon DynamoDB table to be encrypted at rest using an AWS KMS key (p. 450)

[CT.DYNAMODB.PR.1] Require that point-in-time recovery for an Amazon DynamoDB table is activated

This control checks whether point-in-time recovery (PITR) is enabled for an Amazon DynamoDB table.

- **Control objective:** Improve resiliency
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::DynamoDB::Table
- **AWS CloudFormation guard rule:** [CT.DYNAMODB.PR.1 rule specification](p. 447)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.DYNAMODB.PR.1 rule specification](p. 447)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.DYNAMODB.PR.1 example templates](p. 449)

Explanation

Backups help you to recover more quickly from a security incident. They also strengthen the resilience of your systems. Amazon DynamoDB point-in-time recovery (PITR) automates backups for DynamoDB tables, which can reduce the time required to recover from accidental delete or write operations. DynamoDB tables that have PITR enabled can be restored to any point in time within the last 35 days.

Remediation for rule failure

Provide a `PointInTimeRecoverySpecification` configuration and set `PointInTimeRecoveryEnabled` to `true`.
The examples that follow show how to implement this remediation.

**Amazon DynamoDB Table - Example**

Amazon DynamoDB table configured with point-in-time recovery activated. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "DynamoDBTable": {
        "Type": "AWS::DynamoDB::Table",
        "Properties": {
            "AttributeDefinitions": [
                {
                    "AttributeName": "PK",
                    "AttributeType": "S"
                }
            ],
            "BillingMode": "PAY_PER_REQUEST",
            "KeySchema": [
                {
                    "AttributeName": "PK",
                    "KeyType": "HASH"
                }
            ],
            "PointInTimeRecoverySpecification": {
                "PointInTimeRecoveryEnabled": true
            }
        }
    }
}
```

**YAML example**

```
DynamoDBTable:
  Type: AWS::DynamoDB::Table
  Properties:
    AttributeDefinitions:
      - AttributeName: PK
        AttributeType: S
    BillingMode: PAY_PER_REQUEST
    KeySchema:
      - AttributeName: PK
        KeyType: HASH
    PointInTimeRecoverySpecification:
      PointInTimeRecoveryEnabled: true
```

**CT.DYNAMODB.PR.1 rule specification**

```
# ####################################################################
##       Rule Specification       ##
# ####################################################################
# Rule Identifier:
#   dynamodb_table_pitr_enabled_check

```


# Description:
#   This control checks whether point-in-time recovery (PITR) is enabled for an Amazon DynamoDB table.
#
# Reports on:
#   AWS::DynamoDB::Table
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario 1:
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#      And: The input document does not contain any DynamoDB table resources
#      Then: SKIP
#   Scenario 2:
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#      And: The input document does not contain a DynamoDB table resource
#      And: 'PointInTimeRecoverySpecification' is not present on the DynamoDB table resource
#      Then: FAIL
#   Scenario 3:
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#      And: The input document does not contain a DynamoDB table resource
#      And: 'PointInTimeRecoverySpecification' is present on the DynamoDB table resource
#      And: 'PointInTimeRecoveryEnabled' in 'PointInTimeRecoverySpecification' is missing or is a value other than bool(true)
#      Then: FAIL
#   Scenario 4:
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#      And: The input document does not contain a DynamoDB table resource
#      And: 'PointInTimeRecoverySpecification' is present on the DynamoDB table resource
#      And: 'PointInTimeRecoveryEnabled' in 'PointInTimeRecoverySpecification' is present and set to bool(true)
#      Then: PASS
#
# Constants
#
# let DYNAMODB_TABLE_TYPE = "AWS::DynamoDB::Table"
# let INPUT_DOCUMENT = this
#
# Assignments
#
# let dynamodb_tables = Resources.*[ Type == %DYNAMODB_TABLE_TYPE ]
#
# Primary Rules
#
# rule dynamodb_table_pitr_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
#    %dynamodb_tables not empty {
#      check(%dynamodb_tables.Properties)  
#      [<_DYNAMODB.PR.1_]: Require that point-in-time recovery for an Amazon DynamoDB table is activated
#      [<_FIX_]: Provide a 'PointInTimeRecoverySpecification' configuration and set 'PointInTimeRecoveryEnabled' to 'true'.
        }
>>}

rule dynamodb_table_pitr_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %DYNAMODB_TABLE_TYPE) {
  check(%INPUT_DOCUMENT.%DYNAMODB_TABLE_TYPE.resourceProperties)
<<
[CT.DYNAMODB.PR.1]: Require that point-in-time recovery for an Amazon DynamoDB table is activated
[FIX]: Provide a 'PointInTimeRecoverySpecification' configuration and set 'PointInTimeRecoveryEnabled' to 'true'.
>>
}

rule check(dynamodb_table) {
  %dynamodb_table {
    # Scenario 2
    PointInTimeRecoverySpecification exists
    PointInTimeRecoverySpecification is_struct
    # Scenario 3 and 4
    PointInTimeRecoverySpecification {
      PointInTimeRecoveryEnabled exists
      PointInTimeRecoveryEnabled == true
    }
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or Resources exists
  }
}

rule is_cfn_hook(doc, DYNAMODB_TABLE_TYPE) {
  %doc.%DYNAMODB_TABLE_TYPE.resourceProperties exists
}

---

CT.DYNAMODB.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
DynamoDBTable:
  Type: AWS::DynamoDB::Table
  Properties:
    AttributeDefinitions:
      - AttributeName: "PK"
        AttributeType: "S"
    BillingMode: "PAY_PER_REQUEST"
    KeySchema:
      - AttributeName: "PK"
        KeyType: "HASH"
    PointInTimeRecoverySpecification:
      PointInTimeRecoveryEnabled: true
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:

DynamoDBTable:
  Type: AWS::DynamoDB::Table
  Properties:
    AttributeDefinitions:
      - AttributeName: "PK"
        AttributeType: "S"
    BillingMode: "PAY_PER_REQUEST"
    KeySchema:
      - AttributeName: "PK"
        KeyType: "HASH"
    PointInTimeRecoverySpecification:
      PointInTimeRecoveryEnabled: false

[CT.DYNAMODB.PR.2] Require an Amazon DynamoDB table to be encrypted at rest using an AWS KMS key

This control checks whether your Amazon DynamoDB table is encrypted with an AWS Key Management Service (KMS) key.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::DynamoDB::Table
- **AWS CloudFormation guard rule:** CT.DYNAMODB.PR.2 rule specification (p. 451)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.DYNAMODB.PR.2 rule specification (p. 451)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.DYNAMODB.PR.2 example templates (p. 453)

Explanation

Amazon DynamoDB encryption at rest provides an additional layer of data protection, because it always secures your data in an encrypted table - including its primary key, local and global secondary indexes, streams, global tables, backups, and DynamoDB Accelerator (DAX) clusters, whenever the data is stored in durable media.

Encryption at rest integrates with AWS KMS for managing the encryption keys that are used to encrypt your tables.

**Usage considerations**

- This control requires only that KMS keys are used for server-side encryption. It does not check the properties of the KMS key used, such as whether the KMS key is customer-managed or service-managed.
Remediation for rule failure

Provide a SSESpecification configuration and set SSEEnabled to true.

The examples that follow show how to implement this remediation.

Amazon DynamoDB Table - Example

An Amazon DynamoDB table configured to encrypt data at rest with AWS Key Management Service (KMS) keys. The example is shown in JSON and in YAML.

JSON example

```json
{
   "DynamoDBTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
         "AttributeDefinitions": [
            {
               "AttributeName": "PK",
               "AttributeType": "S"
            }
         ],
         "BillingMode": "PAY_PER_REQUEST",
         "KeySchema": [
            {
               "AttributeName": "PK",
               "KeyType": "HASH"
            }
         ],
         "SSESpecification": {
            "SSEEnabled": true
         }
      }
   }
}
```

YAML example

```
DynamoDBTable:
  Type: AWS::DynamoDB::Table
  Properties:
    AttributeDefinitions:
      - AttributeName: PK
        AttributeType: S
    BillingMode: PAY_PER_REQUEST
    KeySchema:
      - AttributeName: PK
        KeyType: HASH
    SSESpecification:
      SSEEnabled: true
```

CT.DYNAMODB.PR.2 rule specification

```
# #################################################################
##       Rule Specification       ##
```

451
Proactive controls

# Rule Identifier:
# dynamodb_table_encrypted_kms_check
#
# Description:
# This control checks whether your Amazon DynamoDB table is encrypted with an AWS Key Management Service (KMS) key.
#
# Reports on:
# AWS::DynamoDB::Table
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain any DynamoDB table resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a DynamoDB table resources
# And: 'SSEEnabled' in 'SSESpecification' has not been provided
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a DynamoDB table resources
# And: 'SSEEnabled' in 'SSESpecification' has been provided and set to a value other than bool(true)
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a DynamoDB table resources
# And: 'SSEEnabled' in 'SSESpecification' has been provided and set to bool(true)
# Then: PASS
#
# Constants
#
let DYNAMODB_TABLE_TYPE = "AWS::DynamoDB::Table"
let INPUT_DOCUMENT = this
#
# Assignments
#
let dynamodb_tables = Resources.*[ Type == %DYNAMODB_TABLE_TYPE ]
#
# Primary Rules
#
rule dynamodb_table_encrypted_kms_check when is_cfn_template(%INPUT_DOCUMENT)
  %dynamodb_tables not empty {

    check(%dynamodb_tables.Properties)
    [CT.DYNAMODB.PR.2]: Require an Amazon DynamoDB table to be encrypted at rest using an AWS KMS key
    [FIX]: Provide a 'SSESpecification' configuration and set 'SSEEnabled' to 'true'.

    <<
    "[CT.DYNAMODB.PR.2]: Require an Amazon DynamoDB table to be encrypted at rest using an AWS KMS key"
    [FIX]: Provide a 'SSESpecification' configuration and set 'SSEEnabled' to 'true'.
    >>
rule dynamodb_table_encrypted_kms_check when is_cfn_hook(%INPUT_DOCUMENT, %DYNAMODB_TABLE_TYPE) {
    check(%INPUT_DOCUMENT.%DYNAMODB_TABLE_TYPE.resourceProperties)
    <<
    [CT.DYNAMODB.PR.2]: Require an Amazon DynamoDB table to be encrypted at rest using an AWS KMS key
    [FIX]: Provide a 'SSESpecification' configuration and set 'SSEEnabled' to 'true'.
    >>
}

rule check(dynamodb_table) {
    %dynamodb_table {
        # Scenario 2
        SSESpecification exists
        SSESpecification is_struct
        # Scenarios 3 and 4
        SSESpecification {
        SSEEnabled exists
        SSEEnabled == true
        }
    }
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or Resources exists
    }
}

rule is_cfn_hook(doc, DYNAMODB_TABLE_TYPE) {
    %doc.%DYNAMODB_TABLE_TYPE.resourceProperties exists
}

CT.DYNAMODB.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
DynamoDBTable:
    Type: AWS::DynamoDB::Table
    Properties:
        AttributeDefinitions:
        - AttributeName: PK
        AttributeType: S
        BillingMode: PAY_PER_REQUEST
        KeySchema:
        - AttributeName: PK
        KeyType: HASH
        SSESpecification:
        SSEEnabled: true
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```json
Resources:
  DynamoDBTable:
    Type: AWS::DynamoDB::Table
    Properties:
      AttributeDefinitions:
        - AttributeName: PK
          AttributeType: S
      BillingMode: PAY_PER_REQUEST
      KeySchema:
        - AttributeName: PK
          KeyType: HASH
      SSESpecification:
        SSEEnabled: false
```

DynamoDB Accelerator controls

Topics

- [CT.DAX.PR.1] Require encryption at rest for all Amazon DynamoDB Accelerator (DAX) clusters (p. 454)
- [CT.DAX.PR.2] Require an Amazon DAX cluster to deploy nodes to at least three Availability Zones (p. 459)

[CT.DAX.PR.1] Require encryption at rest for all Amazon DynamoDB Accelerator (DAX) clusters

This control checks whether Amazon DynamoDB Accelerator (DAX) clusters are encrypted at rest.

Note
The control CT.DAX.PR.1 cannot be activated from home Regions Canada (Central) Region, Europe (Stockholm) Region, and Asia Pacific (Seoul) Region, because the AWS::DAX::Cluster resource type is not available in those Regions. If your home Region is not one of these three, you can activate the control for these three Regions from another home Region, if these three Regions are governed by AWS Control Tower in your landing zone. For example, if your home Region is US West (Oregon) Region, you can deploy the control to Canada (Central) Region, if Canada (Central) Region is governed by AWS Control Tower.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::DAX::Cluster
- **AWS CloudFormation guard rule:** [CT.DAX.PR.1 rule specification](p. 455)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.DAX.PR.1 rule specification](p. 455)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.DAX.PR.1 example templates](p. 457)

Explanation
Encrypting data at rest reduces the risk that data stored on disk may be accessible to a user who is not authenticated to AWS. Encryption adds another set of access controls, which limits the ability of unauthorized users to gain access to the data. For example, API permissions must decrypt the data before it can be read.

**Remediation for rule failure**

Provide an SSESpecification configuration with SSEEnabled set to true.

The examples that follow show how to implement this remediation.

**Amazon DAX Cluster - Example**

Amazon DAX cluster configured with server-side encryption enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "DAXCluster": {
      "Type": "AWS::DAX::Cluster",
      "Properties": {
         "IAMRoleARN": {
            "Fn::GetAtt": [
               "DAXServiceRole",
               "Arn"
            ]
         },
         "NodeType": "dax.t3.small",
         "ReplicationFactor": 1,
         "SSESpecification": {
            "SSEEnabled": true
         }
      }
   }
}
```

**YAML example**

```yaml
DAXCluster:
   Type: AWS::DAX::Cluster
   Properties:
      IAMRoleARN: !GetAtt 'DAXServiceRole.Arn'
      NodeType: dax.t3.small
      ReplicationFactor: 1
      SSESpecification:
         SSEEnabled: true
```

**CT.DAX.PR.1 rule specification**

```bash
# ###################################
##       Rule Specification        ##
#####################################
#
# Rule Identifier:
#   dax_cluster_encryption_enabled_check
#
```


# Description:
# This control checks whether Amazon DynamoDB Accelerator (DAX) clusters are encrypted at rest.
# 
# Reports on:
# AWS::DAX::Cluster
# 
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
# 
# Rule Parameters:
# None
# 
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain any DAX Cluster resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains at least one DAX Cluster resource
# And: 'SSESpecification' has not been provided
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains at least one DAX Cluster resource
# And: 'SSESpecification' has been provided and 'SSESpecification.SSEEnabled' is missing or has been set to a
# value other than bool(true)
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains at least one DAX Cluster resource
# And: 'SSESpecification' has been provided and 'SSESpecification.SSEEnabled' is present and has been set to
# bool(true)
# Then: PASS
#
# Constants
#
let DAX_CLUSTER_TYPE = "AWS::DAX::Cluster"
let INPUT_DOCUMENT = this
#
# Assignments
#
let dax_clusters = Resources.*[ Type == %DAX_CLUSTER_TYPE ]
#
# Primary Rules
#
rule dax_cluster_encryption_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
  %dax_clusters not empty {
    check(%dax_clusters.Properties)
    <<
    [CT.DAX.PR.1]: Require encryption at rest for all Amazon DynamoDB Accelerator (DAX) clusters
    clusters
    [FIX]: Provide an 'SSESpecification' configuration with 'SSEEnabled' set to 'true'.
    >>
}
rule dax_cluster_encryption_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %DAX_CLUSTER_TYPE) {
    check(%INPUT_DOCUMENT.%DAX_CLUSTER_TYPE.resourceProperties)
    <<
    [CT.DAX.PR.1]: Require encryption at rest for all Amazon DynamoDB Accelerator (DAX) 
    clusters
    [FIX]: Provide an 'SSESpecification' configuration with 'SSEEnabled' set to
    'true'.
    >>
}

# Parameterized Rules
#
rule check(dax_cluster) {
    %dax_cluster {
        # Scenario 2
        SSESpecification exists
        SSESpecification is_struct

        # Scenario 3 and 4
        SSESpecification {
            SSEEnabled exists
            SSEEnabled == true
        }
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists  or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.DAX.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
DAXServiceRole:
    Type: AWS::IAM::Role
    Properties:
        AssumeRolePolicyDocument:
            Version: '2012-10-17'
            Statement:
                - Effect: Allow
                    Principal:
                        Service: dax.amazonaws.com
                    Action: sts:AssumeRole
                    Path: /
                    Policies:
- PolicyName: DynamoAccessPolicy
  PolicyDocument:
  Version: '2012-10-17'
  Statement:
  - Effect: Allow
    Action:
    - dynamodb:DescribeTable
    - dynamodb:PutItem
    - dynamodb:GetItem
    - dynamodb:UpdateItem
    - dynamodb:DeleteItem
    - dynamodb:Query
    - dynamodb:Scan
    - dynamodb:BatchGetItem
    - dynamodb:BatchWriteItem
    - dynamodb:ConditionCheckItem
  Resource:
  Fn::Sub: arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:*

DAXCluster:
  Type: AWS::DAX::Cluster
  Properties:
  IAMRoleARN:
    Fn::GetAtt: [DAXServiceRole, Arn]
  NodeType: dax.t3.small
  ReplicationFactor: 1
  SSESpecification:
    SSEEnabled: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
DAXServiceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
      - Effect: Allow
        Principal:
          Service: dax.amazonaws.com
        Action: sts:AssumeRole
        Path: /
      Policies:
      - PolicyName: DynamoAccessPolicy
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
          - Effect: Allow
            Action:
            - dynamodb:DescribeTable
            - dynamodb:PutItem
            - dynamodb:GetItem
            - dynamodb:UpdateItem
            - dynamodb:DeleteItem
            - dynamodb:Query
            - dynamodb:Scan
            - dynamodb:BatchGetItem
            - dynamodb:BatchWriteItem
            - dynamodb:ConditionCheckItem
          Resource:
            Fn::Sub: arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:*
[CT.DAX.PR.2] Require an Amazon DAX cluster to deploy nodes to at least three Availability Zones

This control checks whether an Amazon DAX cluster is configured to deploy cluster nodes to at least three Availability Zones.

- **Control objective**: Improve resiliency, Improve availability
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::DAX::Cluster
- **AWS CloudFormation guard rule**: CT.DAX.PR.2 rule specification (p. 460)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.DAX.PR.2 rule specification (p. 460)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.DAX.PR.2 example templates (p. 463)

**Explanation**

AWS Control Tower recommends that you deploy your Amazon DAX clusters in multiple Availability Zones. This deployment technique allows you to design and operate applications and databases that fail over between Availability Zones automatically, without interruption. For production usage, we strongly recommend that you deploy DAX across at least three nodes, with each node placed into a different Availability Zone.

**Remediation for rule failure**

Set the `ReplicationFactor` parameter to an integer value greater than or equal to three (>= 3), and set the `AvailabilityZones` parameter to a list containing three unique Availability Zone entries.

The examples that follow show how to implement this remediation.

**Amazon DAX cluster - Example**

Amazon DAX cluster configured with three nodes and three distinct Availability Zones. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "DaxCluster": {
        "Type": "AWS::DAX::Cluster",
        "Properties": {
            "IAMRoleARN": {
```

```json
```
"Fn::GetAtt": [
   "DaxDynamoAccessRole",
   "Arn"
],
"NodeType": "dax.t3.small",
"ReplicationFactor": 3,
"AvailabilityZones": [
   {
      "Fn::Select": [
         0,
         {
            "Fn::GetAZs": ""
         }
      },
      
      {
         "Fn::Select": [
            1,
            {
            "Fn::GetAZs": ""
            }
        }
      },
      {
         "Fn::Select": [
            2,
            {
                "Fn::GetAZs": ""
            }
        }
      }
   ]
}

**YAML example**

```yaml
DaxCluster:
  Type: AWS::DAX::Cluster
  Properties:
    IAMRoleARN: !GetAtt 'DaxDynamoAccessRole.Arn'
    NodeType: dax.t3.small
    ReplicationFactor: 3
    AvailabilityZones:
      - !Select
        - 0
        - !GetAZs ''
      - !Select
        - 1
        - !GetAZs ''
      - !Select
        - 2
        - !GetAZs ''
```

**CT.DAX.PR.2 rule specification**
# Rule Specification

## Rule Identifier:
- dax_cluster_multi_az_check

## Description:
- This control checks whether an Amazon DAX cluster is configured to deploy cluster nodes to at least three Availability Zones.

## Reports on:
- AWS::DAX::Cluster

## Evaluates:
- AWS CloudFormation, AWS CloudFormation hook

## Rule Parameters:
- None

## Scenarios:

### Scenario: 1
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document does not contain any DAX cluster resources
  - Then: SKIP

### Scenario: 2
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document contains a DAX cluster resource
  - And: 'ReplicationFactor' has not been provided or has been provided as an integer value less than three (< 3)
  - And: 'AvailabilityZones' has not been provided or provided as an empty list or list with less than three unique entries
  - Then: FAIL

### Scenario: 3
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document contains a DAX cluster resource
  - And: 'ReplicationFactor' has been provided as an integer value greater than or equal to three (>= 3)
  - And: 'AvailabilityZones' has not been provided or provided as an empty list or list with less than three unique entries
  - Then: FAIL

### Scenario: 4
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document contains a DAX cluster resource
  - And: 'ReplicationFactor' has not been provided or has been provided as an integer value less than three (< 3)
  - And: 'AvailabilityZones' has been provided as a list with three or more unique entries
  - Then: FAIL

### Scenario: 5
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document contains a DAX cluster resource
  - And: 'ReplicationFactor' has been provided as an integer value greater than or equal to three (>= 3)
  - And: 'AvailabilityZones' has been provided as a list with three or more unique entries
  - Then: PASS

## Constants

let DAX_CLUSTER_TYPE = "AWS::DAX::Cluster"
let MINIMUM_NODE_COUNT = 3
let INPUT_DOCUMENT = this

# Assignments
let dax_clusters = Resources.*[ Type == %DAX_CLUSTER_TYPE ]

# Primary Rules
rule dax_cluster_multi_az_check when is_cfn_template(%INPUT_DOCUMENT) {
  check(%dax_clusters.Properties)
  %dax_clusters not empty {
    [CT.DAX.PR.2]: Require an Amazon DAX cluster to deploy nodes to at least three Availability Zones
    [FIX]: Set the 'ReplicationFactor' parameter to an integer value greater than or equal to three (>= 3), and set the 'AvailabilityZones' parameter to a list containing three unique Availability Zone entries.
  }
}
rule dax_cluster_multi_az_check when is_cfn_hook(%INPUT_DOCUMENT, %DAX_CLUSTER_TYPE) {
  check(%INPUT_DOCUMENT.%DAX_CLUSTER_TYPE.resourceProperties)
  %INPUT_DOCUMENT.%DAX_CLUSTER_TYPE.resourceProperties not empty {
    [CT.DAX.PR.2]: Require an Amazon DAX cluster to deploy nodes to at least three Availability Zones
    [FIX]: Set the 'ReplicationFactor' parameter to an integer value greater than or equal to three (>= 3), and set the 'AvailabilityZones' parameter to a list containing three unique Availability Zone entries.
  }
}

# Parameterized Rules
rule check(dax_cluster) {
  %dax_cluster {
    # Scenario 2
    ReplicationFactor exists
    AvailabilityZones exists
    AvailabilityZones is_list
    AvailabilityZones not empty

    # Scenarios 3, 4 and 5
    ReplicationFactor >= %MINIMUM_NODE_COUNT
    AvailabilityZones[0] exists
    AvailabilityZones[1] exists
    AvailabilityZones[2] exists
    let az_one = AvailabilityZones[0]
    let az_two = AvailabilityZones[1]
    let az_three = AvailabilityZones[2]
    check_az_is_unique(%az_one, %az_two, %az_three)
    check_az_is_unique(%az_two, %az_one, %az_three)
  }
}
rule check_az_is_unique(az, first_az, second_az) {
  %az not in %first_az
  %az not in %second_az
}
You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
DaxDynamoAccessRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: dax.amazonaws.com
          Action: sts:AssumeRole
          Path: /
          Policies:
            - PolicyName: DynamoAccessPolicy
              PolicyDocument:
                Version: '2012-10-17'
                Statement:
                  - Effect: Allow
                    Action:
                      - dynamodb:DescribeTable
                      - dynamodb:PutItem
                      - dynamodb:GetItem
                      - dynamodb:UpdateItem
                      - dynamodb:DeleteItem
                      - dynamodb:Query
                      - dynamodb:Scan
                      - dynamodb:BatchGetItem
                      - dynamodb:BatchWriteItem
                      - dynamodb:ConditionCheckItem
                  Resource:
                   Fn::Sub: arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:*

DaxCluster:
  Type: AWS::DAX::Cluster
  Properties:
    IAMRoleARN:
      Fn::GetAtt:
        - DaxDynamoAccessRole
        - Arn
    NodeType: dax.t3.small
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
DaxDynamoAccessRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: dax.amazonaws.com
          Action: sts:AssumeRole
          Path: /
        Policies:
          - PolicyName: DynamoAccessPolicy
            PolicyDocument:
              Version: '2012-10-17'
              Statement:
                - Effect: Allow
                  Action:
                    - dynamodb:DescribeTable
                    - dynamodb:PutItem
                    - dynamodb:GetItem
                    - dynamodb:UpdateItem
                    - dynamodb:DeleteItem
                    - dynamodb:Query
                    - dynamodb:Scan
                    - dynamodb:BatchGetItem
                    - dynamodb:BatchWriteItem
                    - dynamodb:ConditionCheckItem
                  Resource:
                    Fn::Sub: arn:${AWS::Partition}:dynamodb:${AWS::Region}:${AWS::AccountId}:*

DaxCluster:
  Type: AWS::DAX::Cluster
  Properties:
    IAMRoleARN:
      Fn::GetAtt:
        - DaxDynamoAccessRole
        - Arn
    NodeType: dax.t3.small
    ReplicationFactor: 3

AWS Elastic Beanstalk controls

Topics
Proactive controls

- **[CT.ELASTICBEANSTALK.PR.1]** Require AWS Elastic Beanstalk environments to have enhanced health reporting enabled (p. 465)
- **[CT.ELASTICBEANSTALK.PR.2]** Require an AWS Elastic Beanstalk environment to have managed platform updates configured (p. 474)
- **[CT.ELASTICBEANSTALK.PR.3]** Require an AWS Elastic Beanstalk environment to have a logging configuration (p. 484)

**[CT.ELASTICBEANSTALK.PR.1]** Require AWS Elastic Beanstalk environments to have enhanced health reporting enabled

This control checks whether AWS Elastic Beanstalk environments and configuration templates are configured for enhanced health reporting.

- **Control objective:** Improve resiliency
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElasticBeanstalk::Environment, AWS::ElasticBeanstalk::ConfigurationTemplate
- **AWS CloudFormation guard rule:** [CT.ELASTICBEANSTALK.PR.1 rule specification](p. 468)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICBEANSTALK.PR.1 rule specification](p. 468)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICBEANSTALK.PR.1 example templates](p. 473)

Explanation

Elastic Beanstalk enhanced health reporting enables a more rapid response to changes in the health of the underlying infrastructure. These changes could result in a lack of availability of the application.

Elastic Beanstalk enhanced health reporting provides a status descriptor to gauge the severity of the identified issues and identify possible causes to investigate. The Elastic Beanstalk health agent, included in supported Amazon Machine Images (AMIs), evaluates logs and metrics of environment EC2 instances.

Remediation for rule failure

For AWS Elastic Beanstalk environments, configure an OptionSetting with Namespace set to aws:elasticbeanstalk:healthreporting:system, OptionName set to SystemType, and Value set to enhanced. For AWS Elastic Beanstalk configuration templates, configure an OptionSetting with Namespace set to aws:elasticbeanstalk:healthreporting:system, OptionName set to SystemType, and Value set to enhanced. Omit this setting to adopt the default value of enhanced.

The examples that follow show how to implement this remediation.

**AWS Elastic Beanstalk Environment - Example**

AWS Elastic Beanstalk environment configured with enhanced health reporting enabled. The example is shown in JSON and in YAML.

**JSON example**
{ "ElasticBeanstalkEnvironment": {
  "Type": "AWS::ElasticBeanstalk::Environment",
  "Properties": {
    "ApplicationName": {
      "Ref": "App"
    },
    "SolutionStackName": "64bit Amazon Linux 2 v3.4.0 running Python 3.8",
    "OptionSettings": [
      {
        "Namespace": "aws:elasticbeanstalk:healthreporting:system",
        "OptionName": "SystemType",
        "Value": "enhanced"
      },
      {
        "Namespace": "aws:autoscaling:launchconfiguration",
        "OptionName": "IamInstanceProfile",
        "Value": {
          "Ref": "InstanceProfile"
        }
      }
    ]
  }
}

YAML example

ElasticBeanstalkEnvironment:
  Type: AWS::ElasticBeanstalk::Environment
  Properties:
    ApplicationName: !Ref 'App'
    SolutionStackName: 64bit Amazon Linux 2 v3.4.0 running Python 3.8
  OptionSettings:
    - Namespace: aws:elasticbeanstalk:healthreporting:system
      OptionName: SystemType
      Value: enhanced
    - Namespace: aws:autoscaling:launchconfiguration
      OptionName: IamInstanceProfile
      Value: !Ref 'InstanceProfile'

The examples that follow show how to implement this remediation.

AWS Elastic Beanstalk Configuration Template - Example One

AWS Elastic Beanstalk configuration template configured with enhanced health reporting, enabled by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.

JSON example

{ "ElasticBeanstalkConfigurationTemplate": {
  "Type": "AWS::ElasticBeanstalk::ConfigurationTemplate",
  "Properties": {
    "ApplicationName": {
      "Ref": "App"
    },
    "SolutionStackName": "64bit Amazon Linux 2 v3.4.0 running Python 3.8",
  }
}
"OptionSettings": [
  {
    "Namespace": "aws:autoscaling:launchconfiguration",
    "OptionName": "IamInstanceProfile",
    "Value": {
      "Ref": "InstanceProfile"
    }
  }
]

YAML example

ElasticBeanstalkConfigurationTemplate:
  Type: AWS::ElasticBeanstalk::ConfigurationTemplate
  Properties:
    ApplicationName: !Ref 'App'
    SolutionStackName: 64bit Amazon Linux 2 v3.4.0 running Python 3.8
    OptionSettings:
      - Namespace: aws:autoscaling:launchconfiguration
        OptionName: IamInstanceProfile
        Value: !Ref 'InstanceProfile'

The examples that follow show how to implement this remediation.

AWS Elastic Beanstalk Configuration Template - Example Two

AWS Elastic Beanstalk configuration template configured with enhanced health reporting, enabled by means of an entry in the OptionSettings property. The example is shown in JSON and in YAML.

JSON example

{
  "ElasticBeanstalkConfigurationTemplate": {
    "Type": "AWS::ElasticBeanstalk::ConfigurationTemplate",
    "Properties": {
      "ApplicationName": {
        "Ref": "App"
      },
      "SolutionStackName": "64bit Amazon Linux 2 v3.4.0 running Python 3.8",
      "OptionSettings": [
        {
          "Namespace": "aws:elasticbeanstalk:healthreporting:system",
          "OptionName": "SystemType",
          "Value": "enhanced"
        }
      ]
    }
  }
}

YAML example
ElasticBeanstalkConfigurationTemplate:
  Type: AWS::ElasticBeanstalk::ConfigurationTemplate
  Properties:
    ApplicationName: !Ref 'App'
    SolutionStackName: 64bit Amazon Linux 2 v3.4.0 running Python 3.8
    OptionSettings:
      - Namespace: aws:elasticbeanstalk:healthreporting:system
        OptionName: SystemType
        Value: enhanced

CT.ELASTICBEANSTALK.PR.1 rule specification

# ####################################################
# Rule Specification
# ####################################################
#
# Rule Identifier:
#   elastic_beanstalk_enhanced_health_reporting_enabled_check
#
# Description:
#   This control checks whether AWS Elastic Beanstalk environments and configuration
#   templates are configured for 'enhanced' health reporting.
#
# Reports on:
#   AWS::ElasticBeanstalk::Environment, AWS::ElasticBeanstalk::ConfigurationTemplate
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#       And: The input document does not contain any Elastic Beanstalk environment
#            resources or
#       Then: SKIP
#   Scenario: 2
#     Given: The input document contains an Elastic Beanstalk environment resource
#       And: 'OptionSettings' is not present in the resource properties or is an empty list
#       Then: FAIL
#   Scenario: 3
#     Given: The input document contains an Elastic Beanstalk environment resource
#       And: 'OptionSettings' is present in the resource properties and is not an empty list
#       Then: FAIL
#       And: No entry in the 'OptionSettings' list has both a 'Namespace' property with a
#            value of 'aws:elasticbeanstalk:healthreporting:system' and an 'OptionName' property with value of 'SystemType'
#       Then: FAIL
#   Scenario: 4
#     Given: The input document contains an Elastic Beanstalk environment resource or an
#            Elastic Beanstalk
### Scenario: 5

Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elastic Beanstalk configuration template resource
And: 'OptionSettings' is present in the resource properties as a non-empty list
And: An entry in the 'OptionSettings' list has a 'Namespace' property with a value of
'aws:elasticbeanstalk:healthreporting:system'
And: That same entry has an 'OptionName' property with a value of 'SystemType'
And: That same entry has a 'Value' property with a value of anything other than 'enhanced', or the 'Value' property is not provided.
Then: FAIL

### Scenario: 6

Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elastic Beanstalk configuration template resource
And: 'OptionSettings' is present in the resource properties as a non-empty list
And: No entry in the 'OptionSettings' list has both a 'Namespace' property with a value of
'aws:elasticbeanstalk:healthreporting:system'
And: an 'OptionName' property with value of 'SystemType'
Then: PASS

### Scenario: 7

Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elastic Beanstalk environment resource or an Elastic Beanstalk configuration template resource
And: 'OptionSettings' is present in the resource properties as a non-empty list
And: Every entry in the 'OptionSettings' list that has both a 'Namespace' property with a value of
'aws:elasticbeanstalk:healthreporting:system'
And: an 'OptionName' property with a value of 'SystemType' also has a 'Value' property with a value of 'enhanced'
Then: PASS

### Constants

```plaintext
let ELASTIC_BEANSTALK_ENVIRONMENT_TYPE = "AWS::ElasticBeanstalk::Environment"
let ELASTIC_BEANSTALK_CONFIGURATION_TEMPLATE_TYPE = "AWS::ElasticBeanstalk::ConfigurationTemplate"
let ELASTIC_BEANSTALK_ENHANCED_HEALTH_REPORTING_NAMESPACE = "aws:elasticbeanstalk:healthreporting:system"
let ELASTIC_BEANSTALK_SYSTEM_TYPE_OPTION_NAME = "SystemType"
let ELASTIC_BEANSTALK_ENHANCED_VALUE = "enhanced"
let INPUT_DOCUMENT = this
```

### Assignments

```plaintext
let elastic_beanstalk_environments = Resources.*[ Type == %ELASTIC_BEANSTALK_ENVIRONMENT_TYPE ]
let elastic_beanstalk_configuration_templates = Resources.*[ Type == %ELASTIC_BEANSTALK_CONFIGURATION_TEMPLATE_TYPE ]
```

### Primary Rules

---

469
rule elastic_beanstalk_enhanced_health_reporting_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%elastic_beanstalk_environments not empty {
    check_elastic_beanstalk_environments(%elastic_beanstalk_environments.Properties)
    <<
        [CT.ELASTICBEANSTALK.PR.1]: Require AWS Elastic Beanstalk environments to have enhanced health reporting enabled
        [FIX]: For AWS Elastic Beanstalk environments, configure an 'OptionSetting' with 'Namespace' set to 'aws:elasticbeanstalk:healthreporting:system', 'OptionName' set to 'SystemType', and 'Value' set to 'enhanced'. For AWS Elastic Beanstalk configuration templates, configure an 'OptionSetting' with 'Namespace' set to 'aws:elasticbeanstalk:healthreporting:system', 'OptionName' set to 'SystemType', and 'Value' set to 'enhanced'. Omit this setting to adopt the default value of 'enhanced'.
    >>}

rule elastic_beanstalk_enhanced_health_reporting_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%elastic_beanstalk_configuration_templates not empty {
    check_elastic_beanstalk_configuration_templates(%elastic_beanstalk_configuration_templates.Properties)
    <<
        [CT.ELASTICBEANSTALK.PR.1]: Require AWS Elastic Beanstalk environments to have enhanced health reporting enabled
        [FIX]: For AWS Elastic Beanstalk environments, configure an 'OptionSetting' with 'Namespace' set to 'aws:elasticbeanstalk:healthreporting:system', 'OptionName' set to 'SystemType', and 'Value' set to 'enhanced'. For AWS Elastic Beanstalk configuration templates, configure an 'OptionSetting' with 'Namespace' set to 'aws:elasticbeanstalk:healthreporting:system', 'OptionName' set to 'SystemType', and 'Value' set to 'enhanced'. Omit this setting to adopt the default value of 'enhanced'.
    >>}

rule elastic_beanstalk_enhanced_health_reporting_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_BEANSTALK_ENVIRONMENT_TYPE) {
    check_elastic_beanstalk_environments(%INPUT_DOCUMENT.%ELASTIC_BEANSTALK_ENVIRONMENT_TYPE.resourceProperties)
    <<
        [CT.ELASTICBEANSTALK.PR.1]: Require AWS Elastic Beanstalk environments to have enhanced health reporting enabled
        [FIX]: For AWS Elastic Beanstalk environments, configure an 'OptionSetting' with 'Namespace' set to 'aws:elasticbeanstalk:healthreporting:system', 'OptionName' set to 'SystemType', and 'Value' set to 'enhanced'. For AWS Elastic Beanstalk configuration templates, configure an 'OptionSetting' with 'Namespace' set to 'aws:elasticbeanstalk:healthreporting:system', 'OptionName' set to 'SystemType', and 'Value' set to 'enhanced'. Omit this setting to adopt the default value of 'enhanced'.
    >>}

rule elastic_beanstalk_enhanced_health_reporting_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_BEANSTALK_CONFIGURATION_TEMPLATE_TYPE) {
    check_elastic_beanstalk_configuration_templates(%INPUT_DOCUMENT.%ELASTIC_BEANSTALK_CONFIGURATION_TEMPLATE_TYPE.resourceProperties)
    <<
        [CT.ELASTICBEANSTALK.PR.1]: Require AWS Elastic Beanstalk environments to have enhanced health reporting enabled
        [FIX]: For AWS Elastic Beanstalk environments, configure an 'OptionSetting' with 'Namespace' set to 'aws:elasticbeanstalk:healthreporting:system', 'OptionName' set to 'SystemType', and 'Value' set to 'enhanced'. For AWS Elastic Beanstalk configuration templates, configure an 'OptionSetting' with 'Namespace' set to 'aws:elasticbeanstalk:healthreporting:system', 'OptionName' set to 'SystemType', and 'Value' set to 'enhanced'. Omit this setting to adopt the default value of 'enhanced'.
    >>}
# Parameterized Rules

## Rule: check_elastic_beanstalk_environments
```
rule check_elastic_beanstalk_environments(elastic_beanstalk_environments) {
    %elastic_beanstalk_environments {
        # Scenario 2
        check_option_settings_exists_or_is_non_empty_list(this)
        # Scenario 3, 4, 7
        check_option_settings_enhanced(OptionSettings[*])
    }
}
```

## Rule: check_elastic_beanstalk_configuration_templates
```
rule check_elastic_beanstalk_configuration_templates(elastic_beanstalk_configuration_templates) {
    %elastic_beanstalk_configuration_templates {
        # Scenario 7
        check_option_settings_with_enhanced_health_reporting(this) or
        # Scenario 6
        check_option_settings_without_health_reporting(this) or
        # Scenario 5
        check_option_settings_not_exists_or_is_empty_list(this)
    }
}
```

## Rule: check_option_settings_with_enhanced_health_reporting
```
rule check_option_settings_with_enhanced_health_reporting(elastic_beanstalk_configuration_templates) {
    %elastic_beanstalk_configuration_templates [
        filter_option_settings_with_health_reporting(this)
    ] {
        check_option_settings_enhanced(OptionSettings[*])
    }
}
```

## Rule: filter_option_settings_with_health_reporting
```
rule filter_option_settings_with_health_reporting(elastic_beanstalk_configuration_templates) {
    some %elastic_beanstalk_configuration_templates {
        check_option_settings_exists_or_is_non_empty_list(this)
        some OptionSettings[*] {
            Namespace exists
            OptionName exists
            Namespace == %ELASTIC_BEANSTALK_ENHANCED_HEALTH_REPORTING_NAMESPACE
            OptionName == %ELASTIC_BEANSTALK_SYSTEM_TYPE_OPTION_NAME
        }
    }
}
```

## Rule: check_option_settings_enhanced
```
rule check_option_settings_enhanced(option_settings) {
    # Scenario 3, 4
    some %option_settings[*] {
        Namespace exists
        OptionName exists
        Value exists
        Namespace == %ELASTIC_BEANSTALK_ENHANCED_HEALTH_REPORTING_NAMESPACE
        OptionName == %ELASTIC_BEANSTALK_SYSTEM_TYPE_OPTION_NAME
        Value == %ELASTIC_BEANSTALK_ENHANCED_VALUE
    }
    # Scenario 7
    let option_setting_duplicates = OptionSettings [
rule check_option_settings_without_health_reporting(elastic_beanstalk_configuration_templates) {
  some %elastic_beanstalk_configuration_templates {
    check_option_settings_exists_or_is_non_empty_list(this)
    let option_settings_with_health_reporting = OptionSettings [
      Namespace exists
      OptionName exists
      Namespace == %ELASTIC_BEANSTALK_ENHANCED_HEALTH_REPORTING_NAMESPACE
      OptionName == %ELASTIC_BEANSTALK_SYSTEM_TYPE_OPTION_NAME
    ]
    %option_settings_with_health_reporting empty
  }
}

rule check_option_settings_exists_or_is_non_empty_list(elastic_beanstalk_resource) {
  %elastic_beanstalk_resource {
    OptionSettings exists
    OptionSettings is_list
    OptionSettings not empty
  }
}

rule check_option_settings_not_exists_or_is_empty_list(configuration_template) {
  %configuration_template { OptionSettings not exists or
    check_is_empty_list(OptionSettings) }
}

rule check_is_empty_list(option_settings) {
  %option_settings {
    this is_list
    this empty
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.ELASTICBEANSTALK.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  InstanceRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
        - Effect: Allow
          Principal:
            Service:
            - elasticbeanstalk.amazonaws.com
          Action:
          - 'sts:AssumeRole'
  InstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Roles:
      - Ref: InstanceRole
  App:
    Type: AWS::ElasticBeanstalk::Application
    ElasticBeanstalkConfigurationTemplate:
      Type: AWS::ElasticBeanstalk::ConfigurationTemplate
      Properties:
        ApplicationName:
        Ref: App
        SolutionStackName: "64bit Amazon Linux 2 v3.4.0 running Python 3.8"
        OptionSettings:
        - Namespace: aws:autoscaling:launchconfiguration
          OptionName: IamInstanceProfile
          Value:
          Ref: InstanceProfile

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  InstanceRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
        - Effect: Allow
          Principal:
            Service:
            - ec2.amazonaws.com
          Action:
          - 'sts:AssumeRole'
  InstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
[CT.ELASTICBEANSTALK.PR.2] Require an AWS Elastic Beanstalk environment to have managed platform updates configured

This control checks whether managed platform updates in AWS Elastic Beanstalk environments and configuration templates are activated.

- **Control objective:** Manage vulnerabilities
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElasticBeanstalk::Environment, AWS::ElasticBeanstalk::ConfigurationTemplate
- **AWS CloudFormation guard rule:** [CT.ELASTICBEANSTALK.PR.2 rule specification (p. 478)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICBEANSTALK.PR.2 rule specification (p. 478)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICBEANSTALK.PR.2 example templates (p. 482)]

**Explanation**

Managed platform updates ensure that the most recent platform fixes, updates, and features for the environment are installed. Keeping patch installations up to date is an important step in securing systems.

**Usage considerations**

- When you set up managed actions on AWS Elastic Beanstalk environments and configuration templates, you must provide PreferredStartTime and UpdateLevel option settings also.
- This control allows you to set up managed actions on AWS Elastic Beanstalk environments only, because environment-level settings take precedence over settings that are defined in configuration templates.
- This control does not allow you to deactivate managed actions on AWS Elastic Beanstalk configuration templates.
Remediation for rule failure

For AWS Elastic Beanstalk environments, create an OptionSetting with a Namespace value set to `aws:elasticbeanstalk:managedactions`, OptionName set to `ManagedActionsEnabled`, and Value set to true. For Elastic Beanstalk configuration templates, create an OptionSetting with a Namespace value set to `aws:elasticbeanstalk:managedactions`, OptionName set to `ManagedActionsEnabled`, and Value set to true, or omit this setting to adopt the default value of true.

The examples that follow show how to implement this remediation.

AWS Elastic Beanstalk Environment - Example

AWS Elastic Beanstalk environment configured with managed platform updates activated. The example is shown in JSON and in YAML.

**JSON example**

```json
{"ElasticBeanstalkEnvironment": {
  "Type": "AWS::ElasticBeanstalk::Environment",
  "Properties": {
    "SolutionStackName": "64bit Amazon Linux 2 v3.4.0 running Python 3.8",
    "ApplicationName": {
      "Ref": "App"
    },
    "OptionSettings": [
      {
        "Namespace": "aws:autoscaling:launchconfiguration",
        "OptionName": "IamInstanceProfile",
        "Value": {
          "Ref": "InstanceProfile"
        }
      },
      {
        "Namespace": "aws:elasticbeanstalk:managedactions",
        "OptionName": "ManagedActionsEnabled",
        "Value": true
      },
      {
        "Namespace": "aws:elasticbeanstalk:managedactions",
        "OptionName": "PreferredStartTime",
        "Value": "Tue:09:00"
      },
      {
        "Namespace": "aws:elasticbeanstalk:managedactions",
        "OptionName": "ServiceRoleForManagedUpdates",
        "Value": "AWSServiceRoleForElasticBeanstalkManagedUpdates"
      },
      {
        "Namespace": "aws:elasticbeanstalk:managedactions:platformupdate",
        "OptionName": "UpdateLevel",
        "Value": "patch"
      }
    ]
  }
}
```

**YAML example**

```yaml
---
ElasticBeanstalkEnvironment: |
  Type: AWS::ElasticBeanstalk::Environment,
  Properties: |
    SolutionStackName: 64bit Amazon Linux 2 v3.4.0 running Python 3.8,
    ApplicationName: |
      Ref: App,
    OptionSettings: |
      - |
        Namespace: aws:autoscaling:launchconfiguration,
        OptionName: IamInstanceProfile,
        Value: |
          Ref: InstanceProfile,
      - |
        Namespace: aws:elasticbeanstalk:managedactions,
        OptionName: ManagedActionsEnabled,
        Value: true,
      - |
        Namespace: aws:elasticbeanstalk:managedactions,
        OptionName: PreferredStartTime,
        Value: Tue:09:00,
      - |
        Namespace: aws:elasticbeanstalk:managedactions,
        OptionName: ServiceRoleForManagedUpdates,
        Value: AWSServiceRoleForElasticBeanstalkManagedUpdates,
      - |
        Namespace: aws:elasticbeanstalk:managedactions:platformupdate,
        OptionName: UpdateLevel,
        Value: patch
---
```
ElasticBeanstalkEnvironment:
  Type: AWS::ElasticBeanstalk::Environment
  Properties:
    SolutionStackName: 64bit Amazon Linux 2 v3.4.0 running Python 3.8
    ApplicationName: !Ref 'App'
  OptionSettings:
    - Namespace: aws:autoscaling:launchconfiguration
      OptionName: IamInstanceProfile
      Value: !Ref 'InstanceProfile'
    - Namespace: aws:elasticbeanstalk:managedactions
      OptionName: ManagedActionsEnabled
      Value: true
    - Namespace: aws:elasticbeanstalk:managedactions
      OptionName: PreferredStartTime
      Value: Tue:09:00
    - Namespace: aws:elasticbeanstalk:managedactions
      OptionName: ServiceRoleForManagedUpdates
      Value: AWSServiceRoleForElasticBeanstalkManagedUpdates
    - Namespace: aws:elasticbeanstalk:managedactions:platformupdate
      OptionName: UpdateLevel
      Value: patch

The examples that follow show how to implement this remediation.

AWS Elastic Beanstalk Configuration Template - Example One

AWS Elastic Beanstalk configuration template configured with managed platform updates enabled, by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ElasticBeanstalkConfigurationTemplate": {
    "Type": "AWS::ElasticBeanstalk::ConfigurationTemplate",
    "Properties": {
      "ApplicationName": {
        "Ref": "App"
      },
      "SolutionStackName": "64bit Amazon Linux 2 v3.4.0 running Python 3.8",
      "OptionSettings": [
        {
          "Namespace": "aws:autoscaling:launchconfiguration",
          "OptionName": "IamInstanceProfile",
          "Value": {
            "Ref": "InstanceProfile"
          }
        }
      ]
    }
  }
}
```

**YAML example**

```yaml
ElasticBeanstalkConfigurationTemplate:
  Type: AWS::ElasticBeanstalk::ConfigurationTemplate
  Properties:
    ApplicationName: !Ref 'App'
    SolutionStackName: 64bit Amazon Linux 2 v3.4.0 running Python 3.8
    OptionSettings:
      - Namespace: aws:autoscaling:launchconfiguration
        OptionName: IamInstanceProfile
        Value: !Ref 'InstanceProfile'
```
Properties:
   ApplicationName: !Ref 'App'
   SolutionStackName: 64bit Amazon Linux 2 v3.4.0 running Python 3.8
OptionSettings:
- Namespace: aws:autoscaling:launchconfiguration
  OptionName: IamInstanceProfile
  Value: !Ref 'InstanceProfile'

The examples that follow show how to implement this remediation.

AWS Elastic Beanstalk Configuration Template - Example Two

AWS Elastic Beanstalk configuration template configured with managed platform updates enabled, by means of an entry in the OptionSettings property. The example is shown in JSON and in YAML.

JSON example

```
{
   "ElasticBeanstalkConfigurationTemplate": {
      "Type": "AWS::ElasticBeanstalk::ConfigurationTemplate",
      "Properties": {
         "ApplicationName": {
            "Ref": "App"
         },
         "SolutionStackName": "64bit Amazon Linux 2 v3.4.0 running Python 3.8",
         "OptionSettings": [
            {
               "Namespace": "aws:elasticbeanstalk:managedactions",
               "OptionName": "ManagedActionsEnabled",
               "Value": true
            },
            {
               "Namespace": "aws:elasticbeanstalk:managedactions",
               "OptionName": "PreferredStartTime",
               "Value": "Tue:09:00"
            },
            {
               "Namespace": "aws:elasticbeanstalk:managedactions:platformupdate",
               "OptionName": "UpdateLevel",
               "Value": "minor"
            }
         ]
      }
   }
}
```

YAML example

```
ElasticBeanstalkConfigurationTemplate:
  Type: AWS::ElasticBeanstalk::ConfigurationTemplate
Properties:
   ApplicationName: !Ref 'App'
   SolutionStackName: 64bit Amazon Linux 2 v3.4.0 running Python 3.8
OptionSettings:
- Namespace: aws:elasticbeanstalk:managedactions
  OptionName: ManagedActionsEnabled
  Value: true
- Namespace: aws:elasticbeanstalk:managedactions
```
OptionName: PreferredStartTime
Value: Tue:09:00
- Namespace: aws:elasticbeanstalk:managedactions:platformupdate
  OptionName: UpdateLevel
  Value: minor

CT.ELASTICBEANSTALK.PR.2 rule specification

```plaintext
# ###################################
##       Rule Specification        
###################################

# Rule Identifier:
#   elastic_beanstalk_managed_updates_enabled_check
#
# Description:
#   This control checks whether managed platform updates in AWS Elastic Beanstalk environments and configuration templates are activated.
#
# Reports on:
#   AWS::ElasticBeanstalk::Environment, AWS::ElasticBeanstalk::ConfigurationTemplate
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any ElasticBeanstalk environment resources or ElasticBeanstalk configuration template resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElasticBeanstalk environment resource
#     And: 'OptionSettings' is not present in the resource properties or is an empty list
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElasticBeanstalk environment resource
#     And: 'OptionSettings' is present in the resource properties as a non-empty list
#     And: No entry in the 'OptionSettings' list has both a 'Namespace' property with a value of 'aws:elasticbeanstalk:managedactions' and an 'OptionName' property with a value of 'ManagedActionsEnabled'
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElasticBeanstalk environment resource or an ElasticBeanstalk configuration template resource
#     And: 'OptionSettings' is present in the resource properties as a non-empty list
#     And: An entry in the 'OptionSettings' list has a 'Namespace' property with a value of
```
# A Scenario

Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document

And: The input document contains an ElasticBeanstalk configuration template resource

And: 'OptionSettings' is not present in the resource properties or is an empty list

Then: FAIL

Scenario: 5

Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document

And: The input document contains an ElasticBeanstalk configuration template resource

And: 'OptionSettings' is present in the resource properties as a non-empty list

And: No entry in the 'OptionSettings' list has both a 'Namespace' property with a value of 'aws:elasticbeanstalk:managedactions' and an 'OptionName' property with a value of 'ManagedActionsEnabled'

Then: PASS

Scenario: 6

Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document

And: The input document contains an ElasticBeanstalk configuration template resource

And: 'OptionSettings' is present in the resource properties as a non-empty list

And: No entry in the 'OptionSettings' list has both a 'Namespace' property with a value of 'aws:elasticbeanstalk:managedactions' and an 'OptionName' property with a value of 'ManagedActionsEnabled'

Then: PASS

Scenario: 7

Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document

And: The input document contains an ElasticBeanstalk environment resource or an ElasticBeanstalk configuration template resource

And: 'OptionSettings' is present in the resource properties as a non-empty list

And: Every entry in the 'OptionSettings' list that has both a 'Namespace' property with a value of 'aws:elasticbeanstalk:managedactions' and an 'OptionName' property with a value of 'ManagedActionsEnabled'

Also has a 'Value' property with a value of bool(true)

Then: PASS

# Constants

let ELASTIC_BEANSTALK_ENVIRONMENT_TYPE = "AWS::ElasticBeanstalk::Environment"

let ELASTIC_BEANSTALK_CONFIGURATION_TEMPLATE_TYPE = "AWS::ElasticBeanstalk::ConfigurationTemplate"

let ELASTIC_BEANSTALK_MANAGED_ACTIONS_NAMESPACE = "aws:elasticbeanstalk:managedactions"

let ELASTIC_BEANSTALK_MANAGED_ACTIONS_OPTION_NAME = "ManagedActionsEnabled"

let ELASTIC_BEANSTALK_MANAGED_ACTIONS_ENABLED_VALUE = ["true", true]

let INPUT_DOCUMENT = this

# Assignments

let elastic_beanstalk_environments = Resources.*[ Type == %ELASTIC_BEANSTALK_ENVIRONMENT_TYPE ]

let elastic_beanstalk_configuration_templates = Resources.*[ Type == %ELASTIC_BEANSTALK_CONFIGURATION_TEMPLATE_TYPE ]

# Primary Rules

rule elastic_beanstalk_managed_updates_enabled_check when is_cfn_template(%INPUT_DOCUMENT)

not empty {
    check_elastic_beanstalk_environments(%elastic_beanstalk_environments.Properties)
}
[CT.ELASTICBEANSTALK.PR.2]: Require an AWS Elastic Beanstalk environment to have managed platform updates configured

[FIX]: For AWS Elastic Beanstalk environments, create an 'OptionSetting' with a 'Namespace' value set to 'aws:elasticbeanstalk:managedactions', 'OptionName' set to 'ManagedActionsEnabled', and 'Value' set to 'true'. For Elastic Beanstalk configuration templates, create an 'OptionSetting' with a 'Namespace' value set to 'aws:elasticbeanstalk:managedactions', 'OptionName' set to 'ManagedActionsEnabled', and 'Value' set to 'true', or omit this setting to adopt the default value of 'true'.

rule elastic_beanstalk_managed_updates_enabled_check when is_cfn_template(%INPUT_DOCUMENT) %elastic_beanstalk_configuration_templates not empty {
  check_elastic_beanstalk_configuration_templates(%elastic_beanstalk_configuration_templates.Properties)
  <<
  [CT.ELASTICBEANSTALK.PR.2]: Require an AWS Elastic Beanstalk environment to have managed platform updates configured
  [FIX]: For AWS Elastic Beanstalk environments, create an 'OptionSetting' with a 'Namespace' value set to 'aws:elasticbeanstalk:managedactions', 'OptionName' set to 'ManagedActionsEnabled', and 'Value' set to 'true'. For Elastic Beanstalk configuration templates, create an 'OptionSetting' with a 'Namespace' value set to 'aws:elasticbeanstalk:managedactions', 'OptionName' set to 'ManagedActionsEnabled', and 'Value' set to 'true', or omit this setting to adopt the default value of 'true'.
  >>
}

rule elastic_beanstalk_managed_updates_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_BEANSTALK_ENVIRONMENT_TYPE) {
  check_elastic_beanstalk_environments(%INPUT_DOCUMENT.%ELASTIC_BEANSTALK_ENVIRONMENT_TYPE.resourceProperties)
  <<
  [CT.ELASTICBEANSTALK.PR.2]: Require an AWS Elastic Beanstalk environment to have managed platform updates configured
  [FIX]: For AWS Elastic Beanstalk environments, create an 'OptionSetting' with a 'Namespace' value set to 'aws:elasticbeanstalk:managedactions', 'OptionName' set to 'ManagedActionsEnabled', and 'Value' set to 'true'. For Elastic Beanstalk configuration templates, create an 'OptionSetting' with a 'Namespace' value set to 'aws:elasticbeanstalk:managedactions', 'OptionName' set to 'ManagedActionsEnabled', and 'Value' set to 'true', or omit this setting to adopt the default value of 'true'.
  >>
}

rule elastic_beanstalk_managed_updates_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_BEANSTALK_CONFIGURATION_TEMPLATE_TYPE) {
  check_elastic_beanstalk_configuration_templates(%INPUT_DOCUMENT.%ELASTIC_BEANSTALK_CONFIGURATION_TEMPLATE_TYPE.resourceProperties)
  <<
  [CT.ELASTICBEANSTALK.PR.2]: Require an AWS Elastic Beanstalk environment to have managed platform updates configured
  [FIX]: For AWS Elastic Beanstalk environments, create an 'OptionSetting' with a 'Namespace' value set to 'aws:elasticbeanstalk:managedactions', 'OptionName' set to 'ManagedActionsEnabled', and 'Value' set to 'true'. For Elastic Beanstalk configuration templates, create an 'OptionSetting' with a 'Namespace' value set to 'aws:elasticbeanstalk:managedactions', 'OptionName' set to 'ManagedActionsEnabled', and 'Value' set to 'true', or omit this setting to adopt the default value of 'true'.
  >>
}

# Parameterized Rules
#
rule check_elastic_beanstalk_environments(elastic_beanstalk_environments) {
  %elastic_beanstalk_environments {
    # Scenario 2
check_option_settings_exists_and_is_non_empty_list(this)

# Scenario 3, 4, 7
check_option_settings_managed_actions_enabled(OptionSettings[*])
}
}

rule check_elastic_beanstalk_configuration_templates(elastic_beanstalk_configuration_templates)
{
  %elastic_beanstalk_configuration_templates {
    # Scenario 7
    check_option_settings_with_managed_actions_enabled(this) or
    # Scenario 6
    check_option_settings_without_managed_actions(this) or
    # Scenario 5
    check_option_settings_not_exists_or_is_empty_list(this)
  }
}

rule check_option_settings_with_managed_actions_enabled(elastic_beanstalk_configuration_templates)
{
  %elastic_beanstalk_configuration_templates {
    filter_option_settings_with_managed_actions(this)
    {
      check_option_settings_managed_actions_enabled(OptionSettings[*])
    }
  }
}

rule filter_option_settings_with_managed_actions(elastic_beanstalk_configuration_templates)
{
  some %elastic_beanstalk_configuration_templates {
    check_option_settings_exists_and_is_non_empty_list(this)
    some OptionSettings[*] {
      Namespace exists
      OptionName exists

      Namespace == %ELASTIC_BEANSTALK_MANAGED_ACTIONS_NAMESPACE
      OptionName == %ELASTIC_BEANSTALK_MANAGED_ACTIONS_OPTION_NAME
    }
  }
}

rule check_option_settings_managed_actions_enabled(option_settings) {
  # Scenario 3, 4
  some %option_settings[*] {
    Namespace exists
    OptionName exists
    Value exists

    Namespace == %ELASTIC_BEANSTALK_MANAGED_ACTIONS_NAMESPACE
    OptionName == %ELASTIC_BEANSTALK_MANAGED_ACTIONS_OPTION_NAME
    Value in %ELASTIC_BEANSTALK_MANAGED_ACTIONS_ENABLED_VALUE
  }

  # Scenario 7
  let option_setting_duplicates = OptionSettings {
    Namespace exists
    OptionName exists
    Value exists

    Namespace == %ELASTIC_BEANSTALK_MANAGED_ACTIONS_NAMESPACE
    OptionName == %ELASTIC_BEANSTALK_MANAGED_ACTIONS_OPTION_NAME
    Value not in %ELASTIC_BEANSTALK_MANAGED_ACTIONS_ENABLED_VALUE
  }
}
%option_setting_duplicates empty

rule
check_option_settings_without_managed_actions(elastic_beanstalk_configuration_templates) {
some %elastic_beanstalk_configuration_templates {
    check_option_settings_exists_and_is_non_empty_list(this)
    
    let option_settings_with_managed_actions = OptionSettings [
        Namespace exists
        OptionName exists
        
        Namespace == %ELASTIC_BEANSTALK_MANAGED_ACTIONS_NAMESPACE
        OptionName == %ELASTIC_BEANSTALK_MANAGED_ACTIONS_OPTION_NAME
    ]

    %option_settings_with_managed_actions empty

}
}

rule check_option_settings_exists_and_is_non_empty_list(elastic_beanstalk_resource) {
    %elastic_beanstalk_resource {
        OptionSettings exists
        OptionSettings is_list
        OptionSettings not empty
    }
}

rule check_option_settings_not_exists_or_is_empty_list(configuration_template) {
    %configuration_template {
        OptionSettings not exists or
        check_is_empty_list(OptionSettings)
    }
}

rule check_is_empty_list(option_settings) {
    %option_settings {
        this is_list
        this empty
    }
}

#
# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.ELASTICBEANSTALK.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.
Resources:
  InstanceRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
        - Effect: Allow
          Principal:
            Service:
            - elasticbeanstalk.amazonaws.com
          Action:
          - 'sts:AssumeRole'
  InstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Roles:
        - Ref: InstanceRole
  App:
    Type: AWS::ElasticBeanstalk::Application
    ElasticBeanstalkConfigurationTemplate:
      Type: AWS::ElasticBeanstalk::ConfigurationTemplate
      Properties:
        ApplicationName:
          Ref: App
        SolutionStackName: "64bit Amazon Linux 2 v3.4.0 running Python 3.8"
        OptionSettings:
          - Namespace: aws:autoscaling:launchconfiguration
            OptionName: IamInstanceProfile
            Value:
              Ref: InstanceProfile

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.
**Proactive controls**

OptionSettings:
- Namespace: aws:autoscaling:launchconfiguration
  OptionName: IamInstanceProfile
  Value: Ref: InstanceProfile
- Namespace: aws:elasticbeanstalk:managedactions
  OptionName: ManagedActionsEnabled
  Value: false

**[CT.ELASTICBEANSTALK.PR.3] Require an AWS Elastic Beanstalk environment to have a logging configuration**

This control checks whether an AWS Elastic Beanstalk environment is configured to send logs to Amazon CloudWatch Logs.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElasticBeanstalk::Environment, AWS::ElasticBeanstalk::ConfigurationTemplate
- **AWS CloudFormation guard rule:** [CT.ELASTICBEANSTALK.PR.3 rule specification](p. 486)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICBEANSTALK.PR.3 rule specification](p. 486)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.ELASTICBEANSTALK.PR.3 example templates](p. 491)

**Explanation**

Monitoring is an important part of maintaining the reliability, availability, and performance of your AWS solutions. We recommend that you collect monitoring data from all of the parts of your AWS solution, so that you can debug a multi-point failure, if one occurs. From a security perspective, logging is an important feature to enable for future forensics efforts in the case of a security incident.

**Usage considerations**

- This control requires only enabling logging to Amazon CloudWatch Logs on AWS Elastic Beanstalk environments, because environment level settings take precedence over settings defined in configuration templates.
- This control does not allow explicitly disabling logging to Amazon CloudWatch Logs on AWS Elastic Beanstalk configuration templates.

**Remediation for rule failure**

For AWS Elastic Beanstalk environments, establish an OptionSetting with a Namespace set to aws:elasticbeanstalk:cloudwatch:logs, OptionName set to StreamLogs, and Value set to true. For Elastic Beanstalk configuration templates, establish an OptionSetting with a Namespace set to aws:elasticbeanstalk:cloudwatch:logs, OptionName set to StreamLogs, and Value set to true, or omit this OptionSetting.

The examples that follow show how to implement this remediation.
AWS Elastic Beanstalk Environment - Example

An AWS Elastic Beanstalk environment configured to stream logs to Amazon CloudWatch Logs by means of an entry in the `OptionSettings` property. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ElasticBeanstalkEnvironment": {
        "Type": "AWS::ElasticBeanstalk::Environment",
        "Properties": {
            "ApplicationName": {
                "Ref": "App"
            },
            "SolutionStackName": "64bit Amazon Linux 2 v3.4.1 running Python 3.8",
            "OptionSettings": [
                {
                    "Namespace": "aws:autoscaling:launchconfiguration",
                    "OptionName": "IamInstanceProfile",
                    "Value": {
                        "Ref": "InstanceProfile"
                    }
                },
                {
                    "Namespace": "aws:elasticbeanstalk:cloudwatch:logs",
                    "OptionName": "StreamLogs",
                    "Value": true
                }
            ]
        }
    }
}
```

**YAML example**

```yaml
ElasticBeanstalkEnvironment:
  Type: AWS::ElasticBeanstalk::Environment
  Properties:
    ApplicationName: !Ref 'App'
    SolutionStackName: 64bit Amazon Linux 2 v3.4.1 running Python 3.8
    OptionSettings:
    - Namespace: aws:autoscaling:launchconfiguration
      OptionName: IamInstanceProfile
      Value: !Ref 'InstanceProfile'
    - Namespace: aws:elasticbeanstalk:cloudwatch:logs
      OptionName: StreamLogs
      Value: true
```

The examples that follow show how to implement this remediation.

AWS Elastic Beanstalk Configuration Template - Example

AWS Elastic Beanstalk configuration template configured to stream logs to Amazon CloudWatch Logs by means of an entry in the `OptionSettings` property. The example is shown in JSON and in YAML.

**JSON example**
YAML example

```yaml
ElasticBeanstalkConfigurationTemplate:
  Type: AWS::ElasticBeanstalk::ConfigurationTemplate
  Properties:
    ApplicationName: !Ref 'App'
    SolutionStackName: 64bit Amazon Linux 2 v3.4.1 running Python 3.8
    OptionSettings:
      - Namespace: aws:autoscaling:launchconfiguration
        OptionName: IamInstanceProfile
        Value: !Ref 'InstanceProfile'
      - Namespace: aws:elasticbeanstalk:cloudwatch:logs
        OptionName: StreamLogs
        Value: true
```

CT.ELASTICBEANSTALK.PR.3 rule specification

```
# ######################################################################
##       Rule Specification        
# ######################################################################

# Rule Identifier:
# elastic_beanstalk_logs_to_cloudwatch_check
# Description:
# This control checks whether an AWS Elastic Beanstalk environment is configured to send logs to Amazon CloudWatch Logs.
# Reports on:
# AWS::ElasticBeanstalk::Environment, AWS::ElasticBeanstalk::ConfigurationTemplate
```
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any ElasticBeanstalk environment resources or ElasticBeanstalk configuration template resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElasticBeanstalk environment resource
#     And: 'OptionSettings' is not present in the resource properties or is an empty list
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElasticBeanstalk environment resource
#     And: 'OptionSettings' is present in the resource properties as a non-empty list
#     And: No entry in the 'OptionSettings' list has both a 'Namespace' property with a value of
#          'aws:elasticbeanstalk:cloudwatch:logs' and an 'OptionName' property with a value of 'StreamLogs'
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElasticBeanstalk environment resource or an ElasticBeanstalk configuration template resource
#     And: 'OptionSettings' is present in the resource properties as a non-empty list
#     And: An entry in the 'OptionSettings' list has a 'Namespace' property with a value of
#          'aws:elasticbeanstalk:cloudwatch:logs'
#     And: That same entry has an 'OptionName' property with a value of 'StreamLogs'
#     And: That same entry has a 'Value' property with a value of anything other than bool(true), or the 'Value' property is not provided.
#     Then: FAIL
#   Scenario: 5
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElasticBeanstalk configuration template resource
#     And: 'OptionSettings' is not present in the resource properties or is an empty list
#     Then: PASS
#   Scenario: 6
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElasticBeanstalk configuration template resource
#     And: 'OptionSettings' is present in the resource properties as a non-empty list
#     And: No entry in the 'OptionSettings' list has both a 'Namespace' property with a value of
#          'aws:elasticbeanstalk:cloudwatch:logs' and an 'OptionName' property with a value of 'StreamLogs'
#     Then: PASS
#   Scenario: 7
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticBeanstalk environment resource or an ElasticBeanstalk configuration template resource
# And: 'OptionSettings' is present in the resource properties as a non-empty list
# And: Every entry in the 'OptionSettings' list that has both a 'Namespace' property with a value of 'aws:elasticbeanstalk:cloudwatch:logs' and an 'OptionName' property with a value of 'StreamLogs' also has a 'Value' property with a value of bool(true)
# Then: PASS

# Constants

let ELASTIC_BEANSTALK_ENVIRONMENT_TYPE = "AWS::ElasticBeanstalk::Environment"
let ELASTIC_BEANSTALK_CONFIGURATION_TEMPLATE_TYPE = "AWS::ElasticBeanstalk::ConfigurationTemplate"
let ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_NAMESPACE = "aws:elasticbeanstalk:cloudwatch:logs"
let ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_OPTION_NAME = "StreamLogs"
let ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_ENABLED_VALUE = ["true", true]
let INPUT_DOCUMENT = this

# Assignments

let elastic_beanstalk_environments = Resources.*[ Type == %ELASTIC_BEANSTALK_ENVIRONMENT_TYPE ]
let elastic_beanstalk_configuration_templates = Resources.*[ Type == %ELASTIC_BEANSTALK_CONFIGURATION_TEMPLATE_TYPE ]

# Primary Rules

rule elastic_beanstalk_logs_to_cloudwatch_check when is_cfn_template(%INPUT_DOCUMENT) %elastic_beanstalk_environments not empty {
    check_elastic_beanstalk_environments(%elastic_beanstalk_environments.Properties)
    <<
    [CT.ELASTICBEANSTALK.PR.3]: Require an AWS Elastic Beanstalk environment to have a logging configuration
    [FIX]: For AWS Elastic Beanstalk environments, establish an 'OptionSetting' with a 'Namespace' set to 'aws:elasticbeanstalk:cloudwatch:logs', 'OptionName' set to 'StreamLogs', and 'Value' set to 'true'. For Elastic Beanstalk configuration templates, establish an 'OptionSetting' with a 'Namespace' set to 'aws:elasticbeanstalk:cloudwatch:logs', 'OptionName' set to 'StreamLogs', and 'Value' set to 'true', or omit this 'OptionSetting'.
    >>
}

rule elastic_beanstalk_logs_to_cloudwatch_check when is_cfn_template(%INPUT_DOCUMENT) %elastic_beanstalk_configuration_templates not empty {
    check_elastic_beanstalk_configuration_templates(%elastic_beanstalk_configuration_templates.Properties)
    <<
    [CT.ELASTICBEANSTALK.PR.3]: Require an AWS Elastic Beanstalk environment to have a logging configuration
    [FIX]: For AWS Elastic Beanstalk environments, establish an 'OptionSetting' with a 'Namespace' set to 'aws:elasticbeanstalk:cloudwatch:logs', 'OptionName' set to 'StreamLogs', and 'Value' set to 'true'. For Elastic Beanstalk configuration templates, establish an 'OptionSetting' with a 'Namespace' set to 'aws:elasticbeanstalk:cloudwatch:logs', 'OptionName' set to 'StreamLogs', and 'Value' set to 'true', or omit this 'OptionSetting'.
    >>
}
rule elastic_beanstalk_logs_to_cloudwatch_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_BEANSTALK_ENVIRONMENT_TYPE) {
  check_elastic_beanstalk_environments(%INPUT_DOCUMENT, %ELASTIC_BEANSTALK_ENVIRONMENT_TYPE.resourceProperties)
  <<
  [CT.ELASTICBEANSTALK.PR.3]: Require an AWS Elastic Beanstalk environment to have a logging configuration.
  [FIX]: For AWS Elastic Beanstalk environments, establish an 'OptionSetting' with a 'Namespace' set to 'aws:elasticbeanstalk:cloudwatch:logs', 'OptionName' set to 'StreamLogs', and 'Value' set to 'true'. For Elastic Beanstalk configuration templates, establish an 'OptionSetting' with a 'Namespace' set to 'aws:elasticbeanstalk:cloudwatch:logs', 'OptionName' set to 'StreamLogs', and 'Value' set to 'true', or omit this 'OptionSetting'.
  >>
}

rule elastic_beanstalk_logs_to_cloudwatch_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_BEANSTALK_CONFIGURATION_TEMPLATE_TYPE) {
  check_elastic_beanstalk_configuration_templates(%INPUT_DOCUMENT, %ELASTIC_BEANSTALK_CONFIGURATION_TEMPLATE_TYPE.resourceProperties)
  <<
  [CT.ELASTICBEANSTALK.PR.3]: Require an AWS Elastic Beanstalk environment to have a logging configuration.
  [FIX]: For AWS Elastic Beanstalk environments, establish an 'OptionSetting' with a 'Namespace' set to 'aws:elasticbeanstalk:cloudwatch:logs', 'OptionName' set to 'StreamLogs', and 'Value' set to 'true'. For Elastic Beanstalk configuration templates, establish an 'OptionSetting' with a 'Namespace' set to 'aws:elasticbeanstalk:cloudwatch:logs', 'OptionName' set to 'StreamLogs', and 'Value' set to 'true', or omit this 'OptionSetting'.
  >>
}

# Parameterized Rules

rule check_elastic_beanstalk_environments(elastic_beanstalk_environments) {
  %elastic_beanstalk_environments {
    # Scenario 2
    check_option_settings_exists_and_is_non_empty_list(this)
    # Scenario 3, 4, 7
    check_option_settings_cloudwatch_logs_enabled(OptionSettings[*])
  }
}

rule check_elastic_beanstalk_configuration_templates(elastic_beanstalk_configuration_templates) {
  %elastic_beanstalk_configuration_templates {
    # Scenario 7
    check_option_settings_with_cloudwatch_logs_enabled(this) or
    # Scenario 6
    check_option_settings_without_cloudwatch_logs(this) or
    # Scenario 5
    check_option_settings_not_exists_or_is_empty_list(this)
  }
}

rule check_option_settings_with_cloudwatch_logs_enabled(elastic_beanstalk_configuration_templates) {
  %elastic_beanstalk_configuration_templates {
    filter_option_settings_with_cloudwatch_logs_enabled(this)
    {
      check_option_settings_cloudwatch_logs_enabled(OptionSettings[*])
    }
  }
}
rule filter_option_settings_with_cloudwatch_logs(elastic_beanstalk_configuration_templates)
{
    some %elastic_beanstalk_configuration_templates 
    check_option_settings_exists_and_is_non_empty_list(this)
    
    some OptionSettings[*] 
    {
        Namespace exists
        OptionName exists
        
        Namespace == %ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_NAMESPACE
        OptionName == %ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_OPTION_NAME
    }
}

rule check_option_settings_cloudwatch_logs_enabled(option_settings) 
{
    # Scenario 3, 4
    some %option_settings[*] 
    {
        Namespace exists
        OptionName exists
        Value exists
        
        Namespace == %ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_NAMESPACE
        OptionName == %ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_OPTION_NAME
        Value in %ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_ENABLED_VALUE
    }

    # Scenario 7
    let option_setting_duplicates = OptionSettings [ 
        Namespace exists
        OptionName exists
        Value exists
        
        Namespace == %ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_NAMESPACE
        OptionName == %ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_OPTION_NAME
        Value not in %ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_ENABLED_VALUE
    ]
    %option_setting_duplicates empty
}

rule check_option_settings_without_cloudwatch_logs(elastic_beanstalk_configuration_templates) 
{
    some %elastic_beanstalk_configuration_templates 
    check_option_settings_exists_and_is_non_empty_list(this)
    
    let option_settings_with_cloudwatch_logs = OptionSettings [ 
        Namespace exists
        OptionName exists
        
        Namespace == %ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_NAMESPACE
        OptionName == %ELASTIC_BEANSTALK_CLOUDWATCH_LOGS_OPTION_NAME
    ]
    %option_settings_with_cloudwatch_logs empty
}

rule check_option_settings_exists_and_is_non_empty_list(elastic_beanstalk_resource) 
{
    %elastic_beanstalk_resource 
    OptionSettings exists
    OptionSettings is_list
    OptionSettings not empty
}
CT.ELASTICBEANSTALK.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  InstanceRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - elasticbeanstalk.amazonaws.com
                - sts:AssumeRole
  InstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Roles:
        - Ref: InstanceRole
  App:
    Type: AWS::ElasticBeanstalk::Application
    Properties: {}
  ElasticBeanstalkEnvironment:
    Type: AWS::ElasticBeanstalk::Environment
    Properties:
      ApplicationName:
        Ref: App
```
PASS Example - Use this template to verify a compliant resource creation.

Resources:
InstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - elasticbeanstalk.amazonaws.com
          Action:
            - sts:AssumeRole
      InstanceProfile:
        Type: AWS::IAM::InstanceProfile
        Properties:
          Roles:
            - Ref: InstanceRole
      App:
        Type: AWS::ElasticBeanstalk::Application
        Properties: {}
    ElasticBeanstalkConfigurationTemplate:
      Type: AWS::ElasticBeanstalk::ConfigurationTemplate
      Properties:
        ApplicationName:
          Ref: App
        SolutionStackName: 64bit Amazon Linux 2 v3.4.1 running Python 3.8
      OptionSettings:
        - Namespace: aws:autoscaling:launchconfiguration
          OptionName: IamInstanceProfile
          Value:
            Ref: InstanceProfile
        - Namespace: aws:elasticbeanstalk:cloudwatch:logs
          OptionName: StreamLogs
          Value: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
InstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - elasticbeanstalk.amazonaws.com
          Action:
            - sts:AssumeRole
Proactive controls

- Effect: Allow
  Principal:
  Service:
  - elasticbeanstalk.amazonaws.com
  Action:
  - sts:AssumeRole
InstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
  Roles:
  - Ref: InstanceRole
App:
  Type: AWS::ElasticBeanstalk::Application
  Properties: {}
ElasticBeanstalkEnvironment:
  Type: AWS::ElasticBeanstalk::Environment
  Properties:
    ApplicationName:
      Ref: App
    SolutionStackName: 64bit Amazon Linux 2 v3.4.1 running Python 3.8
    OptionSettings:
      - Namespace: aws:autoscaling:launchconfiguration
        OptionName: IamInstanceProfile
        Value:
          Ref: InstanceProfile

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
InstanceRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
      - Effect: Allow
        Principal:
          Service:
          - elasticbeanstalk.amazonaws.com
        Action:
        - sts:AssumeRole
InstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
  Roles:
  - Ref: InstanceRole
App:
  Type: AWS::ElasticBeanstalk::Application
  Properties: {}
ElasticBeanstalkConfigurationTemplate:
  Type: AWS::ElasticBeanstalk::ConfigurationTemplate
  Properties:
    ApplicationName:
      Ref: App
    SolutionStackName: 64bit Amazon Linux 2 v3.4.1 running Python 3.8
    OptionSettings:
      - Namespace: aws:autoscaling:launchconfiguration
        OptionName: IamInstanceProfile
        Value:
          Ref: InstanceProfile
      - Namespace: aws:elasticbeanstalk:cloudwatch:logs
        OptionName: StreamLogs
Amazon Elastic Compute Cloud (Amazon EC2) controls

Topics
- [CT.EC2.PR.1] Require an Amazon EC2 launch template to have IMDSv2 configured (p. 494)
- [CT.EC2.PR.2] Require that Amazon EC2 launch templates restrict the token hop limit to a maximum of one (p. 499)
- [CT.EC2.PR.3] Require that any Amazon EC2 security group rule does not use the source IP range 0.0.0.0/0 or ::/0 for ports other than 80 and 443 (p. 504)
- [CT.EC2.PR.4] Require that any Amazon EC2 security group rule does not use the source IP range 0.0.0.0/0 or ::/0 for specific high-risk ports (p. 511)
- [CT.EC2.PR.5] Require any Amazon EC2 network ACL to prevent ingress from 0.0.0.0/0 to port 22 or port 3389 (p. 517)
- [CT.EC2.PR.6] Require that Amazon EC2 transit gateways refuse automatic Amazon VPC attachment requests (p. 524)
- [CT.EC2.PR.7] Require an Amazon EBS volume resource to be encrypted at rest when defined by means of the AWS::EC2::Instance BlockDeviceMappings property or AWS::EC2::Volume resource type (p. 527)
- [CT.EC2.PR.8] Require an Amazon EC2 instance to set AssociatePublicIpAddress to false on a new network interface created by means of the NetworkInterfaces property in the AWS::EC2::Instance resource (p. 534)
- [CT.EC2.PR.9] Require any Amazon EC2 launch template not to auto-assign public IP addresses to network interfaces (p. 540)
- [CT.EC2.PR.10] Require Amazon EC2 launch templates to have Amazon CloudWatch detailed monitoring activated (p. 546)
- [CT.EC2.PR.11] Require that an Amazon EC2 subnet does not automatically assign public IP addresses (p. 549)
- [CT.EC2.PR.12] Require an Amazon EC2 instance to specify at most one network interface by means of the NetworkInterfaces property in the AWS::EC2::Instance resource (p. 554)
- [CT.EC2.PR.13] Require an Amazon EC2 instance to have detailed monitoring enabled (p. 559)
- [CT.EC2.PR.14] Require an Amazon EBS volume configured through an Amazon EC2 launch template to encrypt data at rest (p. 563)

[CT.EC2.PR.1] Require an Amazon EC2 launch template to have IMDSv2 configured

This control checks whether your Amazon EC2 launch templates are configured with Instance Metadata Service Version 2 (IMDSv2).

- **Control objective**: Enforce least privilege, Protect configurations
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::EC2::LaunchTemplate
- **AWS CloudFormation guard rule**: [CT.EC2.PR.1 rule specification](p. 496)

Details and examples
• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.EC2.PR.1 rule specification (p. 496)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.EC2.PR.1 example templates (p. 498)

Explanation

Instance metadata configures and manages your running instances. The IMDS provides access to temporary, frequently rotated credentials, so you don't need to distribute sensitive credentials to instances, either manually or programmatically. The IMDS is attached locally to every EC2 instance. It runs on a special IP address of 169.254.169.254. This IP address is accessible only to software that runs on the instance.

Version 2 of the IMDS adds protections for vulnerabilities that can be used to gain access to the IMDS: Open website application firewalls, Open reverse proxies, Server-side request forgery (SSRF) vulnerabilities and Open Layer 3 firewalls and network address translation (NAT).

AWS Control Tower recommends that you configure your EC2 instances with IMDSv2.

Usage considerations

• This control applies only to Amazon EC2 launch templates that allow access to instance metadata.

Remediation for rule failure

Within the LaunchTemplateData property, provide a MetadataOptions configuration and set the value of HttpTokens to required.

The examples that follow show how to implement this remediation.

Amazon EC2 Launch Template - Example

Amazon EC2 launch template configured with IMDSv2 activated. The example is shown in JSON and in YAML.

JSON example

```
{
  "EC2LaunchTemplate": {  
    "Type": "AWS::EC2::LaunchTemplate",  
    "Properties": {  
      "LaunchTemplateData": {  
        "InstanceType": "t3.micro",  
        "ImageId": {  
          "Ref": "LatestAmiId"  
        },  
        "MetadataOptions": {  
          "HttpTokens": "required"  
        }  
      }  
    }  
  }  
}
```

YAML example

```
---

"EC2LaunchTemplate": {  
  "Type": "AWS::EC2::LaunchTemplate",  
  "Properties": {  
    "LaunchTemplateData": {  
      "InstanceType": "t3.micro",  
      "ImageId": {  
        "Ref": "LatestAmiId"  
      },  
      "MetadataOptions": {  
        "HttpTokens": "required"  
      }  
    }  
  }  
}
```
EC2LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
Properties:
  LaunchTemplateData:
    InstanceType: t3.micro
    ImageId: !Ref 'LatestAmiId'
    MetadataOptions:
      HttpTokens: required

CT.EC2.PR.1 rule specification

# ##################################################################
##       Rule Specification        ##
# ##################################################################
#
# Rule Identifier:
#   ec2_launch_template_imdsv2_check
#
# Description:
#   This control checks whether your Amazon EC2 launch templates are configured with
#   Instance Metadata Service Version 2 (IMDSv2).
#
# Reports on:
#   AWS::EC2::LaunchTemplate
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any EC2 launch template resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EC2 launch template resource
#     And: 'LaunchTemplateData' has not been provided or
#       'LaunchTemplateData.MetadataOptions.HttpEndpoint' has
#       been provided and is equal to 'disabled'
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EC2 launch template resource
#     And: 'LaunchTemplateData' has been provided
#     And: 'MetadataOptions.HttpEndpoint' in 'LaunchTemplateData' has not been provided
#         or has been provided and
#         is equal to 'enabled'
#     And: 'MetadataOptions.HttpTokens' in 'LaunchTemplateData' has not been provided
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EC2 launch template resource
#     And: 'LaunchTemplateData' has been provided
And: 'MetadataOptions.HttpEndpoint' in 'LaunchTemplateData' has not been provided or has been provided and is equal to 'enabled'
And: 'MetadataOptions.HttpTokens' in 'LaunchTemplateData' has been provided and set to a value other than 'required'
Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an EC2 launch template resource
# And: 'LaunchTemplateData' has been provided
# And: 'MetadataOptions.HttpEndpoint' in 'LaunchTemplateData' has not been provided or has been provided and is equal to 'enabled'
# And: 'MetadataOptions.HttpTokens' in 'LaunchTemplateData' has been provided and set to 'required'
# Then: PASS

# Constants

let EC2_LAUNCH_TEMPLATE_TYPE = "AWS::EC2::LaunchTemplate"
let INPUT_DOCUMENT = this

# Assignments

let ec2_launch_templates = Resources.*[ Type == %EC2_LAUNCH_TEMPLATE_TYPE ]

# Primary Rules

rule ec2_launch_template_imdsv2_check when is_cfn_template(%INPUT_DOCUMENT)
%ec2_launch_templates not empty {
  check(%ec2_launch_templates.Properties)
  [CT.EC2.PR.1]: Require an Amazon EC2 launch template to have IMDSv2 configured
  [FIX]: Within the 'LaunchTemplateData' property, provide a 'MetadataOptions' configuration and set the value of 'HttpTokens' to 'required'.
}

rule ec2_launch_template_imdsv2_check when is_cfn_hook(%INPUT_DOCUMENT, %EC2_LAUNCH_TEMPLATE_TYPE) {
  check(%INPUT_DOCUMENT.%EC2_LAUNCH_TEMPLATE_TYPE.resourceProperties)
  [CT.EC2.PR.1]: Require an Amazon EC2 launch template to have IMDSv2 configured
  [FIX]: Within the 'LaunchTemplateData' property, provide a 'MetadataOptions' configuration and set the value of 'HttpTokens' to 'required'.
}

# Parameterized Rules

rule check(launch_template) {
%launch_template [
  # Scenario 2
  filter_launch_template_imdss_enabled(this)
]
  LaunchTemplateData exists
  LaunchTemplateData is_struct
  LaunchTemplateData {
    # Scenario 3, 4 and 5
    MetadataOptions exists
}
MetadataOptions is_struct
MetadataOptions {
    HttpTokens exists
    HttpTokens == "required"
}
}
}

rule filter_launch_template_imds_enabled(launch_template) {
    %launch_template {
        LaunchTemplateData exists
        LaunchTemplateData is_struct
        LaunchTemplateData {
            MetadataOptions not exists or
            filter_metadata_options_imds_enabled(this)
        }
    }
}

rule filter_metadata_options_imds_enabled(metadata_options) {
    %metadata_options {
        MetadataOptions is_struct
        MetadataOptions {
            HttpEndpoint not exists or
            HttpEndpoint == "enabled"
        }
    }
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.EC2.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Parameters:
- LatestAmiId:
  Description: Region specific latest AMI ID from the Parameter Store
  Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
- EC2LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateData:
      InstanceType: t3.micro
      ImageId:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Parameters:
- LatestAmiId:
  - Description: Region specific latest AMI ID from the Parameter Store
  - Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  - Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
- EC2LaunchTemplate:
  - Type: AWS::EC2::LaunchTemplate
  - Properties:
    - LaunchTemplateData:
      - InstanceType: t3.micro
      - ImageId:
        - Ref: LatestAmiId
      - MetadataOptions:
        - HttpTokens: optional

[CT.EC2.PR.2] Require that Amazon EC2 launch templates restrict the token hop limit to a maximum of one

This control checks whether an Amazon EC2 launch template has a metadata token hop limit set to 1.

- **Control objective:** Enforce least privilege, Protect configurations
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EC2::LaunchTemplate
- **AWS CloudFormation guard rule:** [CT.EC2.PR.2 rule specification](p. 501)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.EC2.PR.2 rule specification](p. 501)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.EC2.PR.2 example templates](p. 504)

Explanation

The Amazon Instance Metadata Service (IMDS) provides metadata information about an Amazon EC2 instance, which is useful for application configuration. Restricting the HTTP PUT response for the metadata service to the EC2 instance protects the IMDS from unauthorized use.

The Time To Live (TTL) field in the IP packet is reduced by one on every hop. This reduction can be used to ensure that the packet does not travel outside EC2. IMDSv2 protects EC2 instances that may have been misconfigured as open routers, layer 3 firewalls, VPNS, tunnels, or NAT devices, which prevents unauthorized users from retrieving metadata. With IMDSv2, the PUT response that contains the secret token cannot travel outside the instance, because the default metadata response hop limit is set to 1. However, if this value is greater than 1, the token can leave the EC2 instance.
Usage considerations

- This control applies only to Amazon EC2 launch templates that allow access to instance metadata.
- This control is incompatible with Amazon EC2 launch templates that require a token hop limit of 2.

Remediation for rule failure

Within the LaunchTemplateData property, provide a MetadataOptions configuration with the value of HttpPutResponseLimit set to 1, or omit the HttpPutResponseLimit property to adopt the AWS CloudFormation default value of 1.

The examples that follow show how to implement this remediation.

Amazon EC2 Launch Template - Example One

Amazon EC2 launch template configured with access to instance metadata enabled and a token hop limit of 1, set by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "EC2LaunchTemplate": {
      "Type": "AWS::EC2::LaunchTemplate",
      "Properties": {
         "LaunchTemplateData": {
            "MetadataOptions": {
               "HttpEndpoint": "enabled"
            }
         }
      }
   }
}
```

**YAML example**

```yaml
EC2LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateData:
      MetadataOptions:
        HttpEndpoint: enabled
```

The examples that follow show how to implement this remediation.

Amazon EC2 Launch Template - Example Two

Amazon EC2 launch template configured with access to instance metadata enabled and a token hop limit of 1, set by means of the MetadataOptions property. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "EC2LaunchTemplate": {
```
"Type": "AWS::EC2::LaunchTemplate",
"Properties": {
  "LaunchTemplateData": {
    "MetadataOptions": {
      "HttpEndpoint": "enabled",
      "HttpPutResponseHopLimit": 1
    }
  }
}
}

YAML example

EC2LaunchTemplate:  
  Type: AWS::EC2::LaunchTemplate  
Properties:  
  LaunchTemplateData:  
    MetadataOptions:  
      HttpEndpoint: enabled  
      HttpPutResponseHopLimit: 1

CT.EC2.PR.2 rule specification

# #####################################################################
## Rule Specification     ##
# ########################################################################
# # Rule Identifier:  
# # ec2_launch_template_token_hop_limit_check  
# # Description:  
# #   This control checks whether an Amazon EC2 launch template has a metadata token hop limit set to '1'.  
# # Reports on:  
# #   AWS::EC2::LaunchTemplate  
# # Evaluates:  
# #   AWS CloudFormation, AWS CloudFormation hook  
# # Rule Parameters:  
# #   None  
# # Scenarios:  
# # Scenario: 1  
# #   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document  
# #   And: The input document does not contain any EC2 launch template resources  
# #   Then: SKIP  
# # Scenario: 2  
# #   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document  
# #   And: The input document contains an EC2 launch template resource  
# #   And: 'LaunchTemplateData.MetadataOptions' has been provided  
# #   And: 'LaunchTemplateData.MetadataOptions.HttpEndpoint' has been provided and is equal to 'disabled'  
# #   Then: SKIP  
# # Scenario: 3

501
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EC2 launch template resource
# And: 'LaunchTemplateData.MetadataOptions' has been provided
# And: 'LaunchTemplateData.MetadataOptions.HttpEndpoint' has not been provided or has been provided and is equal to 'enabled'
# And: 'LaunchTemplateData.MetadataOptions.HttpPutResponseHopLimit' has been provided and is not equal to an integer of 1.
# Then: FAIL

# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EC2 launch template resource
# And: 'LaunchTemplateData.MetadataOptions' has not been provided
# Then: PASS

# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EC2 launch template resource
# And: 'LaunchTemplateData.MetadataOptions' has been provided
# And: 'LaunchTemplateData.MetadataOptions.HttpEndpoint' has not been provided or has been provided and is equal to 'enabled'
# And: 'LaunchTemplateData.MetadataOptions.HttpPutResponseHopLimit' has not been provided
# Then: PASS

# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EC2 launch template resource
# And: 'LaunchTemplateData.MetadataOptions' has been provided
# And: 'LaunchTemplateData.MetadataOptions.HttpEndpoint' has not been provided or has been provided and is equal to 'enabled'
# And: 'LaunchTemplateData.MetadataOptions.HttpPutResponseHopLimit' has been provided and is equal to an integer of 1.
# Then: PASS

# Constants
let EC2_LAUNCH_TEMPLATE_TYPE = "AWS::EC2::LaunchTemplate"
let INPUT_DOCUMENT = this

# Assignments
let ec2_launch_templates = Resources.*[ Type == %EC2_LAUNCH_TEMPLATE_TYPE ]

# Primary Rules
# Primary Rules
rule ec2_launch_template_token_hop_limit_check when is_cfn_template(this)
  %ec2_launch_templates not empty {
    check(%ec2_launch_templates.Properties)
    [CT.EC2.PR.2]: Require that Amazon EC2 launch templates restrict the token hop limit to a maximum of one
    [FIX]: Within the 'LaunchTemplateData' property, provide a 'MetadataOptions' configuration with the value of 'HttpPutResponseLimit' set to '1', or omit the 'HttpPutResponseLimit' property to adopt the AWS CloudFormation default value of '1'.
  }

502
rule ec2_launch_template_token_hop_limit_check when is_cfn_hook(%INPUT_DOCUMENT, %EC2_LAUNCH_TEMPLATE_TYPE) {
  check(%INPUT_DOCUMENT.%EC2_LAUNCH_TEMPLATE_TYPE.resourceProperties)
  <<
  [CT.EC2.PR.2]: Require that Amazon EC2 launch templates restrict the token hop limit to a maximum of one
  [FIX]: Within the 'LaunchTemplateData' property, provide a 'MetadataOptions' configuration with the value of 'HttpPutResponseLimit' set to '1', or omit the 'HttpPutResponseLimit' property to adopt the AWS CloudFormation default value of '1'.
  >>
}

# Parameterized Rules
#
rule check(ec2_launch_template) {
  %ec2_launch_template[
    # Scenario 2, 3 and 4
    filter_launch_template(this)
  ] {
    # Scenario 5 and 6
    LaunchTemplateData {
      MetadataOptions not exists or
      MetadataOptions {
        HttpPutResponseHopLimit not exists or
        HttpPutResponseHopLimit == 1
      }
    }
  }
}

rule filter_launch_template(ec2_launch_template) {
  %ec2_launch_template {
    LaunchTemplateData exists
    LaunchTemplateData is_struct
    LaunchTemplateData {
      MetadataOptions not exists or
      filter_metadata_options_provided(this)
    }
  }
}

rule filter_metadata_options_provided(options) {
  %options {
    MetadataOptions is_struct
    MetadataOptions {
      HttpEndpoint not exists or
      HttpEndpoint == "enabled"
    }
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.EC2.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Parameters:
  LatestAmiId:
    Description: Region specific latest AMI ID from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
  EC2LaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateData:
        InstanceType: t3.micro
        ImageId:
          Ref: LatestAmiId

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  EC2LaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateData:
        MetadataOptions:
          HttpPutResponseHopLimit: 2

[CT.EC2.PR.3] Require that any Amazon EC2 security group rule does not use the source IP range `0.0.0.0/0` or `::/0` for ports other than 80 and 443

This control checks whether an Amazon EC2 security group rule contains the string `0.0.0.0/0` or `::/0` as a source IP range. This control is not triggered if a rule allows connection to port 80 or 443 with TCP, UDP, ICMP, or ICMPv6.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EC2::SecurityGroup, AWS::EC2::SecurityGroupIngress
- **AWS CloudFormation guard rule:** [CT.EC2.PR.3 rule specification](p. 506)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.EC2.PR.3 rule specification](p. 506)
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.EC2.PR.3 example templates](p. 509)
**Explanation**

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources. Disallowing usage of the strings 0.0.0.0/0 or ::/0 helps protect against this common misconfiguration and encourages users to choose a range aligned with least-privilege principles.

AWS recommends a layered approach, to ensure that network access is provided only as necessary for your business requirements. Security group rules should follow the principle of least privileged access. Unrestricted access increases the opportunity for malicious activity. Unless a port is specifically allowed, the port should deny unrestricted access (any IP address with a /0 suffix).

**Usage considerations**

- This control applies only to Amazon EC2 security group and EC2 security group ingress resources with ingress rules that allow inbound traffic from 0.0.0.0/0 or ::/0

**Remediation for rule failure**

Ensure that security groups with ingress rules that allow TCP or UDP traffic from 0.0.0.0/0 or ::/0 allow traffic from ports 80 or 443 only.

The examples that follow show how to implement this remediation.

**Amazon EC2 Security Group - Example One**

Amazon EC2 Security Group allowing inbound TCP traffic from 0.0.0.0/0 on port 80. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "SecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": {
        "Fn::Sub": "$\{AWS::StackName\}-example"
      },
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "CidrIp": "0.0.0.0/0",
          "FromPort": 80,
          "ToPort": 80
        }
      ]
    }
  }
}
```

**YAML example**

```yaml
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: !Sub '{AWS::StackName}-example'
    SecurityGroupIngress:
    - IpProtocol: tcp
      CidrIp: '0.0.0.0/0'
      FromPort: 80
```

---

**505**
The examples that follow show how to implement this remediation.

**Amazon EC2 Security Group - Example Two**

Amazon EC2 Security Group allowing inbound TCP traffic from `0.0.0.0/0` on port 443. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "SecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": {
                "Fn::Sub": "${AWS::StackName}-example"
            },
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "CidrIp": "0.0.0.0/0",
                    "FromPort": 443,
                    "ToPort": 443
                }
            ]
        }
    }
}
```

**YAML example**

```
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: !Sub '${AWS::StackName}-example'
    SecurityGroupIngress:
      - IpProtocol: tcp
        CidrIp: '0.0.0.0/0'
        FromPort: 443
        ToPort: 443
```

**CT.EC2.PR.3 rule specification**

```bash
# #######################################################################
##       Rule Specification        
#######################################################################
#
# Rule Identifier:
# vpc_sg_open_only_to_authorized_ports_check
#
# Description:
# This control checks whether the Amazon EC2 security group contains the string
# '0.0.0.0/0' or '::/0' as a source IP range.
```
This control is not triggered if a rule allows connection to port 80 or 443 with TCP, UDP, ICMP, or ICMPv6.

Reports on:
- AWS::EC2::SecurityGroup, AWS::EC2::SecurityGroupIngress

Evaluates:
- AWS CloudFormation, AWS CloudFormation hook

Rule Parameters:
- None

Scenarios:
- Scenario: 1
  Given: The input document is an AWS CloudFormation or CloudFormation hook document
  And: The input document does not contain any Amazon EC2 security group or EC2 security group ingress resources
  Then: SKIP

- Scenario: 2
  Given: The input document is an AWS CloudFormation or CloudFormation hook document
  And: The input document contains an Amazon EC2 security group resource or EC2 security group ingress resource
  Then: SKIP

- Scenario: 3
  Given: The input document is an AWS CloudFormation or CloudFormation hook document
  And: The input document contains an Amazon EC2 security group resource or EC2 security group ingress resource
  Then: SKIP

- Scenario: 4
  Given: The input document is an AWS CloudFormation or CloudFormation hook document
  And: The input document contains an Amazon EC2 security group resource or EC2 security group ingress resource
  Then: FAIL

- Scenario: 5
  Given: The input document is an AWS CloudFormation or CloudFormation hook document
  And: The input document contains an Amazon EC2 security group resource or EC2 security group ingress resource
  Then: PASS
let SECURITY_GROUP_TYPE = "AWS::EC2::SecurityGroup"
let SECURITY_GROUP_INGRESS_TYPE = "AWS::EC2::SecurityGroupIngress"
let ALLOWED_PORTS = [80, 443]
let AUTHORIZED_PROTOCOLS = ["tcp", "udp", "icmp", "icmpv6"]
let UNRESTRICTED_IPV4_RANGES = ["0.0.0.0/0"]
let UNRESTRICTED_IPV6_RANGES = ["::/0"]
let INPUT_DOCUMENT = this

# Assignments
#
let ec2_security_groups = Resources[
    Type == %SECURITY_GROUP_TYPE
]
let ec2_security_group_ingress_rules = Resources[
    Type == %SECURITY_GROUP_INGRESS_TYPE
]

# Primary Rules
#
rule vpc_sg_open_only_to_authorized_ports_check when is_cfn_template(%INPUT_DOCUMENT)
%ec2_security_groups not empty {
    check_security_group(%ec2_security_groups.Properties)
    <<<
    [CT.EC2.PR.3]: Require that any Amazon EC2 security group rule does not use the source IP range 0.0.0.0/0 or ::/0 for ports other than 80 and 443
    [FIX]: Ensure that security groups with ingress rules that allow TCP or UDP traffic from '0.0.0.0/0' or '::/0' only allow traffic to ports 80 or 443. The use of managed prefix lists is not supported.
    >>>
}

rule vpc_sg_open_only_to_authorized_ports_check when is_cfn_template(%INPUT_DOCUMENT)
%ec2_security_group_ingress_rules not empty {
    check_ingress_rule(%ec2_security_group_ingress_rules.Properties)
    <<<
    [CT.EC2.PR.3]: Require that any Amazon EC2 security group rule does not use the source IP range 0.0.0.0/0 or ::/0 for ports other than 80 and 443
    [FIX]: Ensure that security groups with ingress rules that allow TCP or UDP traffic from '0.0.0.0/0' or '::/0' only allow traffic to ports 80 or 443. The use of managed prefix lists is not supported.
    >>>
}

rule vpc_sg_open_only_to_authorized_ports_check when is_cfn_hook(%INPUT_DOCUMENT, %SECURITY_GROUP_TYPE) {
    check_security_group(%INPUT_DOCUMENT.%SECURITY_GROUP_TYPE.resourceProperties)
    <<<
    [CT.EC2.PR.3]: Require that any Amazon EC2 security group rule does not use the source IP range 0.0.0.0/0 or ::/0 for ports other than 80 and 443
    [FIX]: Ensure that security groups with ingress rules that allow TCP or UDP traffic from '0.0.0.0/0' or '::/0' only allow traffic to ports 80 or 443. The use of managed prefix lists is not supported.
    >>>
}

rule vpc_sg_open_only_to_authorized_ports_check when is_cfn_hook(%INPUT_DOCUMENT, %SECURITY_GROUP_INGRESS_TYPE) {
    check_ingress_rule(%INPUT_DOCUMENT.%SECURITY_GROUP_INGRESS_TYPE.resourceProperties)
    <<<

508
[CT.EC2.PR.3]: Require that any Amazon EC2 security group rule does not use the source IP range 0.0.0.0/0 or ::/0 for ports other than 80 and 443.

[FIX]: Ensure that security groups with ingress rules that allow TCP or UDP traffic from '0.0.0.0/0' or '::/0' only allow traffic to ports 80 or 443. The use of managed prefix lists is not supported.

```plaintext
# Parameterized Rules
#
rule check_security_group(security_group) {
  %security_group [SecurityGroupIngress exists
   SecurityGroupIngress is_list
   SecurityGroupIngress not empty
  ] {
    SecurityGroupIngress[*] {
      check_ingress_rule(this)
    }
  }
}

rule check_ingress_rule(ingress_rule) {
  %ingress_rule[ CidrIp in %UNRESTRICTED_IPV4_RANGES or
   CidrIpv6 in %UNRESTRICTED_IPV6_RANGES or
   SourcePrefixListId exists ] {
    # Scenario 3
    IpProtocol exists
    IpProtocol in %AUTHORIZED_PROTOCOLS
    when IpProtocol in ["tcp", "udp"] {
      FromPort exists
      ToPort exists
      # Scenarios 4 and 5
      check_ports(FromPort, ToPort)
    }
  }
}

rule check_ports(from_port, to_port) {
  %from_port in %ALLOWED_PORTS
  %to_port in %ALLOWED_PORTS
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc [AWSTemplateFormatVersion exists or
   Resources exists
  ]
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

CT.EC2.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.
PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription:
        Fn::Sub: ${AWS::StackName}-example
      SecurityGroupIngress:
        - IpProtocol: tcp
          CidrIp: 0.0.0.0/0
          FromPort: 80
          ToPort: 80
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```yaml
Resources:
  PrefixList:
    Type: AWS::EC2::PrefixList
    Properties:
      PrefixListName:
        Fn::Sub: ${AWS::StackName}-example
      AddressFamily: IPv4
      MaxEntries: 10
      Entries:
        - Cidr: "0.0.0.0/0"
          Description: Public internet
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription:
        Fn::Sub: ${AWS::StackName}-example
  SecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId:
        Fn::GetAtt: [ SecurityGroup, GroupId ]
      IpProtocol: -1
      SourcePrefixListId:
        Ref: PrefixList
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```yaml
Resources:
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription:
        Fn::Sub: ${AWS::StackName}-example
  SecurityGroupIngress:
    Type: AWS::EC2::SecurityGroupIngress
    Properties:
      GroupId:
        Fn::GetAtt: [ SecurityGroup, GroupId ]
      IpProtocol: udp
      CidrIp: 0.0.0.0/0
```
FromPort: 80
ToPort: 90

[CT.EC2.PR.4] Require that any Amazon EC2 security group rule does not use the source IP range 0.0.0.0/0 or ::/0 for specific high-risk ports

This control checks whether an Amazon EC2 security group rule that contains the strings 0.0.0.0/0 or ::/0 as a source IP range does not allow incoming TCP, UDP, ICMP, or ICMPv6 traffic to the following ports: 3389, 20, 23, 110, 143, 3306, 8080, 1433, 9200, 9300, 25, 445, 135, 21, 1434, 4333, 5432, 5500, 5601, 22, 3000, 5000, 8088, 8888.

• Control objective: Limit network access
• Implementation: AWS CloudFormation Guard Rule
• Control behavior: Proactive
• Resource types: AWS::EC2::SecurityGroup, AWS::EC2::SecurityGroupIngress
• AWS CloudFormation guard rule: CT.EC2.PR.4 rule specification (p. 513)

Details and examples

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.EC2.PR.4 rule specification (p. 513)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.EC2.PR.4 example templates (p. 516)

Explanation

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources. Disallowing usage of the strings 0.0.0.0/0 or ::/0 helps protect against this common misconfiguration and encourages users to choose a range aligned with least-privilege principles.

AWS recommends a layered approach, to ensure that network access is provided only as necessary for your business requirements. No security group should allow unrestricted ingress access to the following ports:

3389, 20, 23, 110, 143, 3306, 8080, 1433, 9200, 9300, 25, 445, 135, 21, 1434, 4333, 5432, 5500, 5601, 22, 3000, 5000, 8088, 8888.

Unrestricted access (0.0.0.0/0) increases opportunities for malicious activity, such as hacking, denial-of-service attacks, and loss of data.

Usage considerations

• This control applies only to Amazon EC2 security group and security group ingress resources with ingress rules that allow inbound traffic from 0.0.0.0/0 or ::/0.

Remediation for rule failure

Remove Amazon EC2 security group ingress rules that allow traffic from 0.0.0.0/0 or ::/0 to high-risk ports: 3389, 20, 23, 110, 143, 3306, 8080, 1433, 9200, 9300, 25, 445, 135, 21, 1434, 4333, 5432, 5500, 5601, 22, 3000, 5000, 8088, 8888.

The examples that follow show how to implement this remediation.
Amazon EC2 Security Group - Example

Amazon EC2 security group configured to allow traffic from the source IP range 0.0.0.0/0 or ::/0 on a port range that does not include a high-risk port. The example is shown in JSON and in YAML.

**JSON example**

```
{
    "SecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "sample-security-group",
            "SecurityGroupIngress": [
                {
                    "IpProtocol": "tcp",
                    "CidrIp": "0.0.0.0/0",
                    "FromPort": 80,
                    "ToPort": 80
                }
            ]
        }
    }
}
```

**YAML example**

```
SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
        GroupDescription: sample-security-group
        SecurityGroupIngress:
        - IpProtocol: tcp
          CidrIp: '0.0.0.0/0'
          FromPort: 80
          ToPort: 80
```

The examples that follow show how to implement this remediation.

Amazon EC2 Security Group Ingress Rule - Example

Amazon EC2 security group ingress rule configured to allow traffic from the source IP range 0.0.0.0/0 or ::/0 on a port range that does not include a high-risk port. The example is shown in JSON and in YAML.

**JSON example**

```
{
    "SecurityGroupIngress": {
        "Type": "AWS::EC2::SecurityGroupIngress",
        "Properties": {
            "GroupId": {
                "Fn::GetAtt": [
                    "SecurityGroup",
                    "GroupId"
                ]
            },
            "IpProtocol": "tcp",
            "CidrIp": "0.0.0.0/0",
            "FromPort": 80,
            "ToPort": 80
        }
    }
}
```
YAML example

SecurityGroupIngress:
  Type: AWS::EC2::SecurityGroupIngress
  Properties:
    GroupId: !GetAtt 'SecurityGroup.GroupId'
    IpProtocol: tcp
    CidrIp: '0.0.0.0/0'
    FromPort: 80
    ToPort: 90

CT.EC2.PR.4 rule specification

# ##################################################################################################
## Rule Specification
# ##################################################################################################
#
# Rule Identifier:
#   vpc_sg_restricted_common_ports_check
#
# Description:
#   This control checks whether an Amazon EC2 security group rule that contains the strings
#   '0.0.0.0/0' or '::/0' as a source IP range
#   does not allow incoming TCP, UDP, ICMP, ICMPv6 traffic to the following ports: '3389',
#   '20', '23', '110', '143',
#   '3306', '8080', '1433', '9200', '9300', '25', '445', '135', '21', '1434', '4333',
#   '5432', '5500', '5601', '22', '3000', '5000',
#   '8088', '8888'.
#
# Reports on:
#   AWS::EC2::SecurityGroup, AWS::EC2::SecurityGroupIngress
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation Hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any EC2 security group or EC2 security
group ingress resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EC2 security group resource or EC2 security
group ingress resource
#     And: EC2 security group or EC2 security group ingress resource does not allow
inbound traffic from a source
#     prefix list and has no rules allowing inbound traffic from source '0.0.0.0/0'
# or '::/0'
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EC2 security group resource or EC2 security group ingress resource
# And: EC2 security group or EC2 security group ingress resource has rules allowing inbound traffic
#   from a source prefix list, or source '0.0.0.0/0' or '::/0'
# And: EC2 security group or EC2 security group ingress resource has a rule that allows all traffic
#   ('IpProtocol' is set to '-1' or another protocol number)
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EC2 security group resource or EC2 security group ingress resource
# And: EC2 security group or EC2 security group ingress resource has rules allowing inbound traffic
#   from a source prefix list, or source '0.0.0.0/0' or '::/0'
# And: EC2 security group or EC2 security group ingress resource has no rules that allow all traffic
#   ('IpProtocol' is not set to '-1' or another protocol number)
# And: Ports allowed are in the list of blocked ports
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EC2 security group resource or EC2 security group ingress resource
# And: EC2 security group or EC2 security group ingress resource has rules allowing inbound traffic
#   from a source prefix list, or source '0.0.0.0/0' or '::/0'
# And: EC2 security group or EC2 security group ingress resource has no rules that allow all traffic
#   ('IpProtocol' is not set to '-1' or another protocol number)
# And: Ports allowed are not in the list of blocked ports
# Then: PASS

# Constants
let SECURITY_GROUP_TYPE = "AWS::EC2::SecurityGroup"
let SECURITY_GROUP_INGRESS_TYPE = "AWS::EC2::SecurityGroupIngress"
let BLOCKED_PORTS = [3389, 20, 23, 110, 143, 3306, 8080, 1433, 9200, 9300, 25, 445, 135, 21, 1434, 4333, 5432, 5500, 5601, 22, 3000, 5000, 8088, 8888]
let AUTHORIZED_PROTOCOLS = ["tcp", "udp", "icmp", "icmpv6"]
let UNRESTRICTED_IPV4_RANGES = ["0.0.0.0/0"]
let UNRESTRICTED_IPV6_RANGES = ["::/0"]
let INPUT_DOCUMENT = this

# Assignments
let ec2_security_groups = Resources[
    Type == %SECURITY_GROUP_TYPE
]
let ec2_security_group_ingress_rules = Resources[
    Type == %SECURITY_GROUP_INGRESS_TYPE
]

# Primary Rules

rule vpc_sg_restricted_common_ports_check when is_cfn_template(%INPUT_DOCUMENT)
%ec2_security_groups not empty {
  check_security_group(%ec2_security_groups.Properties)
  <<
  [CT.EC2.PR.4]: Require that any Amazon EC2 security group rule does not use the source IP range 0.0.0.0/0 or ::/0 for specific high-risk ports
  [FIX]: Remove Amazon EC2 security group ingress rules that allow traffic from '0.0.0.0/0' or '::/0' to high-risk ports: '3389', '20', '23', '110', '143', '3306', '8000', '1433', '9200', '9300', '25', '445', '135', '21', '1434', '6333', '5432', '5500', '5601', '22', '3000', '5000', '8088', '8888'. The use of managed prefix lists is not supported.
  >>
}

rule vpc_sg_restricted_common_ports_check when is_cfn_template(%INPUT_DOCUMENT)
%ec2_security_group_ingress_rules not empty {
  check_ingress_rule(%ec2_security_group_ingress_rules.Properties)
  <<
  [CT.EC2.PR.4]: Require that any Amazon EC2 security group rule does not use the source IP range 0.0.0.0/0 or ::/0 for specific high-risk ports
  [FIX]: Remove Amazon EC2 security group ingress rules that allow traffic from '0.0.0.0/0' or '::/0' to high-risk ports: '3389', '20', '23', '110', '143', '3306', '8000', '1433', '9200', '9300', '25', '445', '135', '21', '1434', '6333', '5432', '5500', '5601', '22', '3000', '5000', '8088', '8888'. The use of managed prefix lists is not supported.
  >>
}

rule vpc_sg_restricted_common_ports_check when is_cfn_hook(%INPUT_DOCUMENT, %SECURITY_GROUP_TYPE) {
  check_security_group(%INPUT_DOCUMENT.%SECURITY_GROUP_TYPE.resourceProperties)
  <<
  [CT.EC2.PR.4]: Require that any Amazon EC2 security group rule does not use the source IP range 0.0.0.0/0 or ::/0 for specific high-risk ports
  [FIX]: Remove Amazon EC2 security group ingress rules that allow traffic from '0.0.0.0/0' or '::/0' to high-risk ports: '3389', '20', '23', '110', '143', '3306', '8000', '1433', '9200', '9300', '25', '445', '135', '21', '1434', '6333', '5432', '5500', '5601', '22', '3000', '5000', '8088', '8888'. The use of managed prefix lists is not supported.
  >>
}

rule vpc_sg_restricted_common_ports_check when is_cfn_hook(%INPUT_DOCUMENT, %SECURITY_GROUP_INGRESS_TYPE) {
  check_ingress_rule(%INPUT_DOCUMENT.%SECURITY_GROUP_INGRESS_TYPE.resourceProperties)
  <<
  [CT.EC2.PR.4]: Require that any Amazon EC2 security group rule does not use the source IP range 0.0.0.0/0 or ::/0 for specific high-risk ports
  [FIX]: Remove Amazon EC2 security group ingress rules that allow traffic from '0.0.0.0/0' or '::/0' to high-risk ports: '3389', '20', '23', '110', '143', '3306', '8000', '1433', '9200', '9300', '25', '445', '135', '21', '1434', '6333', '5432', '5500', '5601', '22', '3000', '5000', '8088', '8888'. The use of managed prefix lists is not supported.
  >>
}

# Parameterized Rules
# rule check_security_group(security_group) {
  %security_group [
SecurityGroupIngress exists
SecurityGroupIngress is_list
SecurityGroupIngress not empty
} {
  SecurityGroupIngress[*] {
    check_ingress_rule(this)
  }
}
}

rule check_ingress_rule(ingress_rule) {
  %ingress_rule[ CidrIp in %UNRESTRICTED_IPV4_RANGES or
  CidrIpv6 in %UNRESTRICTED_IPV6_RANGES or
  SourcePrefixListId exists ] {
    # Scenario 3
    IpProtocol exists
    IpProtocol in %AUTHORIZED_PROTOCOLS
    
    when IpProtocol in ["tcp", "udp"] {
      FromPort exists
      ToPort exists
      
      let ingress_block = this
      
      %BLOCKED_PORTS.* {
        # Scenarios 4 and 5
        check_ports(this, %ingress_block.FromPort, %ingress_block.ToPort)
      }
    }
  }
}

rule check_ports(port, FromPort, ToPort) {
  %FromPort > %port or
  %ToPort < %port
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.EC2.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription:
      Fn::Sub: ${AWS::StackName}-example
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
PrefixList:
  Type: AWS::EC2::PrefixList
  Properties:
    PrefixListName:
      Fn::Sub: ${AWS::StackName}-example
    AddressFamily: IPv4
    MaxEntries: 10
    Entries:
      - Cidr: "0.0.0.0/0"
        Description: Public internet
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription:
      Fn::Sub: ${AWS::StackName}-example
    SecurityGroupIngress:
      - IpProtocol: -1
        SourcePrefixListId:
          Ref: PrefixList

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription:
      Fn::Sub: ${AWS::StackName}-example
    SecurityGroupIngress:
      - IpProtocol: tcp
        CidrIp: 0.0.0.0/0
        FromPort: 22
        ToPort: 22

[CT.EC2.PR.5] Require any Amazon EC2 network ACL to prevent ingress from 0.0.0.0/0 to port 22 or port 3389

This control checks whether the Amazon EC2 network ACL inbound entry allows unrestricted incoming traffic (0.0.0.0/0 or ::/0) for SSH or RDP.

- **Control objective**: Limit network access
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
**Resource types:** AWS::EC2::NetworkAclEntry

**AWS CloudFormation guard rule:** [CT.EC2.PR.5 rule specification (p. 520)]

### Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: [CT.EC2.PR.5 rule specification (p. 520)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.EC2.PR.5 example templates (p. 523)]

### Explanation

Access to remote server administration ports, such as port 22 (SSH) and port 3389 (RDP), should not be publicly accessible, because these ports may allow unintended access to resources within your VPC.

**Usage considerations**

- This control only applies to Amazon EC2 network ACL entry resources that allow unrestricted inbound traffic.

### Remediation for rule failure

For Amazon EC2 network ACL entries that allow inbound connectivity on port 22 or port 3389, provide a CIDR range in CidrBlock or Ipv6CidrBlock that does not allow traffic from all sources.

The examples that follow show how to implement this remediation.

#### Amazon EC2 Network ACL Entry - Example One

Amazon EC2 network ACL entry configured to allow unrestricted inbound IPv4 TCP traffic in a port range excluding port 22 (SSH) and port 3389 (RDP). The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "NetworkAclEntry": {
        "Type": "AWS::EC2::NetworkAclEntry",
        "Properties": {
            "CidrBlock": "0.0.0.0/0",
            "Egress": false,
            "NetworkAclId": {
                "Ref": "NACL"
            },
            "Protocol": 6,
            "PortRange": {
                "From": 2000,
                "To": 2005
            },
            "RuleAction": "allow",
            "RuleNumber": 100
        }
    }
}
```

**YAML example**

```yaml
- NetworkAclEntry:
  - Type: AWS::EC2::NetworkAclEntry
    Properties:
      CidrBlock: 0.0.0.0/0
      Egress: false
      NetworkAclId:
        Ref: NACL
      Protocol: 6
      PortRange:
        From: 2000
        To: 2005
      RuleAction: allow
      RuleNumber: 100
```
The examples that follow show how to implement this remediation.

**Amazon EC2 Network ACL Entry - Example Two**

Amazon EC2 network ACL entry configured to allow unrestricted inbound IPv6 UDP traffic in a port range excluding port 3389 (RDP). The example is shown in JSON and in YAML.

**JSON example**

```
{
    "NetworkAclEntry": {
        "Type": "AWS::EC2::NetworkAclEntry",
        "Properties": {
            "Ipv6CidrBlock": "::/0",
            "Egress": false,
            "NetworkAclId": {
                "Ref": "NACL"
            },
            "Protocol": 17,
            "PortRange": {
                "From": 100,
                "To": 200
            },
            "RuleAction": "allow",
            "RuleNumber": 100
        }
    }
}
```

**YAML example**

```
NetworkAclEntry:
  Type: AWS::EC2::NetworkAclEntry
  Properties:
    Ipv6CidrBlock: ::/0
    Egress: false
    NetworkAclId: !Ref 'NACL'
    Protocol: 17
    PortRange:
      From: 100
      To: 200
    RuleAction: allow
    RuleNumber: 100
```
## CT.EC2.PR.5 rule specification

<table>
<thead>
<tr>
<th># Rule Identifier:</th>
<th>nacl_no_unrestricted_ssh_rdp_check</th>
</tr>
</thead>
<tbody>
<tr>
<td># Description:</td>
<td>This control checks whether the Amazon EC2 network ACL inbound entry allows unrestricted incoming traffic ('0.0.0.0/0' or '::/0') for SSH or RDP.</td>
</tr>
<tr>
<td># Reports on:</td>
<td>AWS::EC2::NetworkAclEntry</td>
</tr>
<tr>
<td># Evaluates:</td>
<td>AWS CloudFormation, AWS CloudFormation hook</td>
</tr>
<tr>
<td># Rule Parameters:</td>
<td>None</td>
</tr>
<tr>
<td># Scenarios:</td>
<td></td>
</tr>
<tr>
<td># Scenario 1</td>
<td>Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document AND: The input document does not contain any EC2 network ACL entry resources THEN: SKIP</td>
</tr>
<tr>
<td># Scenario 2</td>
<td>Given: The input document contains a EC2 network ACL entry resource AND: EC2 network ACL entry resource has no CIDR block allowing inbound traffic from source '0.0.0.0/0' or '::/0' THEN: SKIP</td>
</tr>
<tr>
<td># Scenario 3</td>
<td>Given: The input document contains a EC2 network ACL entry resource AND: EC2 network ACL entry resource allows inbound traffic from source '0.0.0.0/0' or '::/0' AND: EC2 network ACL entry resource allows all traffic ('IpProtocol' is set to '-1') THEN: FAIL</td>
</tr>
<tr>
<td># Scenario 4</td>
<td>Given: The input document contains a EC2 network ACL entry resource AND: EC2 network ACL entry resource allows inbound traffic from source '0.0.0.0/0' or '::/0' AND: EC2 network ACL entry resource allows TCP (protocol 6) traffic AND: EC2 network ACL entry resource allows traffic from a PortRange that includes 22 THEN: FAIL</td>
</tr>
<tr>
<td># Scenario 5</td>
<td>Given: The input document contains a EC2 network ACL entry resource AND: EC2 network ACL entry resource allows inbound traffic from source '0.0.0.0/0' or '::/0' AND: EC2 network ACL entry resource allows TCP (protocol 6) or UDP (protocol 17) traffic AND: EC2 network ACL entry resource allows traffic from a PortRange that includes 3389 THEN: FAIL</td>
</tr>
</tbody>
</table>
Then: FAIL
Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a EC2 network ACL entry resource
And: EC2 network ACL entry resource allows inbound traffic from source '0.0.0.0/0' or '::/0'
And: EC2 network ACL entry resource allows TCP (protocol 6) traffic
And: EC2 network ACL entry resource allows traffic from a PortRange that excludes 22
Then: PASS
Scenario: 7
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a EC2 network ACL entry resource
And: EC2 network ACL entry resource allows inbound traffic from source '0.0.0.0/0' or '::/0'
And: EC2 network ACL entry resource allows TCP (protocol 6) or UDP (protocol 17) traffic
And: EC2 network ACL entry resource allows traffic from a PortRange that excludes 3389
Then: PASS

Constants

let NETWORK_ACL_TYPE = "AWS::EC2::NetworkAclEntry"
let INPUT_DOCUMENT = this
let ALL_TRAFFIC_PROTOCOL = [-1, "-1"]
let TCP_PROTOCOL = [6, "6"]
let UDP_PROTOCOL = [17, "17"]
let UNRESTRICTED_IPV4_RANGES = ["0.0.0.0/0"]
let UNRESTRICTED_IPV6_RANGES = [":/0"]
let SSH_PORT = 22
let RDP_PORT = 3389

Assignments

let nacl_entries = Resources.*[ Type == %NETWORK_ACL_TYPE ]

Primary Rules

rule nacl_no_unrestricted_ssh_rdp_check when is_cfn_template(%INPUT_DOCUMENT)
%nacl_entries not empty {
  check(%nacl_entries.Properties)
  nacl_entries not empty {
    [CT.EC2.PR.5]: Require any Amazon EC2 network ACL to prevent ingress from 0.0.0.0/0 to port 22 or port 3389
    [FIX]: For Amazon EC2 network ACL entries that allow inbound connectivity on port 22 or port 3389, provide a CIDR range in 'CidrBlock' or 'Ipv6CidrBlock' that does not allow traffic from all sources.
  }
}

rule nacl_no_unrestricted_ssh_rdp_check when is_cfn_hook(%INPUT_DOCUMENT, %NETWORK_ACL_TYPE) {
  check(%INPUT_DOCUMENT.%NETWORK_ACL_TYPE.resourceProperties)
  [CT.EC2.PR.5]: Require any Amazon EC2 network ACL to prevent ingress from 0.0.0.0/0 to port 22 or port 3389
  [FIX]: For Amazon EC2 network ACL entries that allow inbound connectivity on port 22 or port 3389, provide a CIDR range in 'CidrBlock' or 'Ipv6CidrBlock' that does not allow traffic from all sources.
} # # Parameterized Rules
# rule check(nacl_entry) {
%nacl_entry [
    # Scenario 2
    filter_allow_unrestricted_ingress(this)
] {
    # Scenario 3
    Protocol exists
    Protocol not in %ALL_TRAFFIC_PROTOCOL
    # Scenario 4, 6
    check_for_open_ssh(this)
    # Scenario 5, 7
    check_for_open_rdp(this)
}
}
rule filter_allow_unrestricted_ingress(nacl_entry) {
    Egress not exists or
    Egress != true
    CidrBlock in %UNRESTRICTED_IPV4_RANGES or
    Ipv6CidrBlock in %UNRESTRICTED_IPV6_RANGES
    RuleAction == "allow"
}
rule check_for_open_ssh(nacl_entry) {
    %nacl_entry [
    Protocol in %TCP_PROTOCOL
] {
    check_port_range_exists(this)
    check_ports(%SSH_PORT, PortRange.From, PortRange.To)
}
}
rule check_for_open_rdp(nacl_entry) {
    %nacl_entry [
    Protocol in %TCP_PROTOCOL or
    Protocol in %UDP_PROTOCOL
] {
    check_port_range_exists(this)
    check_ports(%RDP_PORT, PortRange.From, PortRange.To)
}
}
rule check_port_range_exists(nacl_entry) {
    PortRange exists
    PortRange is_struct
    PortRange {
    From exists
    To exists
    }
}
rule check_ports(port, nacl_from_port, nacl_to_port) {
    %nacl_from_port > %port or
    %nacl_to_port < %port
}
# Utility Rules

**CT.EC2.PR.5 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```yaml
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 192.168.0.0/16
  NACL:
    Type: AWS::EC2::NetworkAcl
    Properties:
      VpcId:
        Ref: VPC
    NetworkAclEntry:
      Type: AWS::EC2::NetworkAclEntry
      Properties:
        CidrBlock: 0.0.0.0/0
        Egress: false
        NetworkAclId:
          Ref: NACL
        Protocol: 6
        PortRange:
          From: 2000
          To: 2005
        RuleAction: allow
        RuleNumber: 100
```

**FAIL Example** - Use this template to verify that the control prevents non-compliant resource creation.

```yaml
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 192.168.0.0/16
  NACL:
    Type: AWS::EC2::NetworkAcl
    Properties:
      VpcId:
        Ref: VPC
    NetworkAclEntry:
      Type: AWS::EC2::NetworkAclEntry
      Properties:
```

523
[CT.EC2.PR.6] Require that Amazon EC2 transit gateways refuse automatic Amazon VPC attachment requests

This control checks whether Amazon EC2 transit gateways are configured to accept Amazon VPC attachment requests automatically.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EC2::TransitGateway
- **AWS CloudFormation guard rule:** [CT.EC2.PR.6 rule specification (p. 525)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.EC2.PR.6 rule specification (p. 525)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.EC2.PR.6 example templates (p. 527)]

**Explanation**

Turning on the AutoAcceptSharedAttachments property configures a transit gateway to accept cross-account VPC attachment requests automatically, without verifying the request or the account from which the attachment is originating. In alignment with the best practices of authorization and authentication, we recommended turning off this feature, to ensure that only authorized VPC attachment requests are accepted.

**Remediation for rule failure**

Omit the AutoAcceptSharedAttachments property or set the property to disable.

The examples that follow show how to implement this remediation.

**AWS Transit Gateway - Example**

AWS Transit Gateway configured to deactivate auto-acceptance of cross-account Amazon VPC attachments. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "TransitGateway": {
      "Type": "AWS::EC2::TransitGateway",
      "Properties": {
```

```json
524
```
"AutoAcceptSharedAttachments": "disable"
}
}
}

YAML example

TransitGateway:
  Type: AWS::EC2::TransitGateway
  Properties:
    AutoAcceptSharedAttachments: disable

CT.EC2.PR.6 rule specification

# ###################################################################
##       Rule Specification        ##
# ###################################################################
#
# Rule Identifier:
#   ec2_transit_gateway_auto_vpc_attach_disabled_check
#
# Description:
#   This control checks whether Amazon EC2 transit gateways are configured to accept Amazon
#   VPC attachment requests automatically.
#
# Reports on:
#   AWS::EC2::TransitGateway
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any EC2 transit gateway resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EC2 transit gateway resource
#     And: 'AutoAcceptSharedAttachments' configuration has been provided and is set to a
#     value other than 'disable'
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EC2 transit gateway resource
#     And: 'AutoAcceptSharedAttachments' configuration has not been provided
#     Then: PASS
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EC2 transit gateway resource
#     And: 'AutoAcceptSharedAttachments' configuration has been provided and set to
#     'disable'
# Constants

let EC2_TRANSIT_GATEWAY_TYPE = "AWS::EC2::TransitGateway"
let INPUT_DOCUMENT = this

# Assignments

let ec2_transit_gateway = Resources.*[ Type == %EC2_TRANSIT_GATEWAY_TYPE ]

# Primary Rules

rule ec2_transit_gateway_auto_vpc_attach_disabled_check when is_cfn_template(%INPUT_DOCUMENT)
    %ec2_transit_gateway not empty
    {
        check(%ec2_transit_gateway.Properties)
        <<
            [CT.EC2.PR.6]: Require that Amazon EC2 transit gateways refuse automatic Amazon VPC attachment requests
            [FIX]: Omit the 'AutoAcceptSharedAttachments' property or set the property to 'disable'.
        >>
    }

rule ec2_transit_gateway_auto_vpc_attach_disabled_check when is_cfn_hook(%INPUT_DOCUMENT, %EC2_TRANSIT_GATEWAY_TYPE)
    {
        check(%INPUT_DOCUMENT.%EC2_TRANSIT_GATEWAY_TYPE.resourceProperties)
        <<
            [CT.EC2.PR.6]: Require that Amazon EC2 transit gateways refuse automatic Amazon VPC attachment requests
            [FIX]: Omit the 'AutoAcceptSharedAttachments' property or set the property to 'disable'.
        >>
    }

# Parameterized Rules

rule check(ec2_transit_gateway) { %ec2_transit_gateway { # Scenario 3
    AutoAcceptSharedAttachments not exists or
    # Scenario 2 and 4
    AutoAcceptSharedAttachments == "disable"
} }

# Utility Rules

rule is_cfn_template(doc) { %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
} }

rule is_cfn_hook(doc, RESOURCE_TYPE) { %doc.%RESOURCE_TYPE.resourceProperties exists }
CT.EC2.PR.6 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
TransitGateway:
  Type: AWS::EC2::TransitGateway
  Properties:
    AutoAcceptSharedAttachments: disable

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
TransitGateway:
  Type: AWS::EC2::TransitGateway
  Properties:
    AutoAcceptSharedAttachments: enable

[CT.EC2.PR.7] Require an Amazon EBS volume resource to be encrypted at rest when defined by means of the AWS::EC2::Instance BlockDeviceMappings property or AWS::EC2::Volume resource type

This control checks whether your standalone Amazon EC2 EBS volume and Amazon Elastic Block Store (EBS) volume created through EC2 instance Block Device Mappings are encrypted at rest. Specifically, it checks that the Encrypted property is set to true in either the EBS volume resource definition or an EC2 instance resource definition's BlockDeviceMappings property.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EC2::Instance, AWS::EC2::Volume
- **AWS CloudFormation guard rule:** [CT.EC2.PR.7 rule specification (p. 530)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.EC2.PR.7 rule specification (p. 530)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.EC2.PR.7 example templates (p. 532)]

Explanation

For an added layer of security of your sensitive data in Amazon EC2 EBS volumes, you should enable EBS encryption at rest. Amazon EBS encryption offers a straightforward encryption solution for your EBS resources that doesn't require you to build, maintain, and secure your own key management infrastructure. It uses KMS keys when creating encrypted volumes and snapshots.
Amazon Elastic Block Store (EBS) volumes can be inherited from:

- The Amazon Machine Image (AMI) specified with the `ImageId` property
- The Launch Template specified with the `LaunchTemplateId` property

**Usage considerations**

- For Amazon EC2 instance block device mappings, this control does not check any block device mappings created by means of an EC2 launch template or inherited through the AMI from which the instance is launched.

**Remediation for rule failure**

Set Encryption to true on Amazon EC2 EBS Volumes.

The examples that follow show how to implement this remediation.

**Amazon EC2 Instance - Example**

Amazon EC2 instance with an encrypted EBS volume. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "EC2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
      "ImageId": {
        "Ref": "LatestAmiId"
      },
      "InstanceType": "t3.micro",
      "NetworkInterfaces": [
        {
          "DeviceIndex": 0,
          "SubnetId": {
            "Ref": "Subnet"
          },
          "AssociatePublicIpAddress": false
        }
      ],
      "BlockDeviceMappings": [
        {
          "DeviceName": "/dev/sdm",
          "Ebs": {
            "VolumeType": "gp3",
            "Iops": 200,
            "Encrypted": true,
            "DeleteOnTermination": true,
            "VolumeSize": 20
          }
        }
      ]
    }
  }
}
```

**YAML example**

```yaml
resources:
  - type: AWS::EC2::Instance
    properties:
      ImageId: ${LatestAmiId}
      InstanceType: t3.micro
      NetworkInterfaces:
      - DeviceIndex: 0
        SubnetId: ${Subnet}
        AssociatePublicIpAddress: false
      BlockDeviceMappings:
      - DeviceName: /dev/sdm
        Ebs:
          VolumeType: gp3
          Iops: 200
          Encrypted: true
          DeleteOnTermination: true
          VolumeSize: 20
```
The examples that follow show how to implement this remediation.

Amazon EBS Volume - Example

Amazon EBS Volume with encryption configured. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "EBSVolume": {
      "Type": "AWS::EC2::Volume",
      "Properties": {
         "Size": 100,
         "AvailabilityZone": {
            "Fn::Select": [0,
            {
               "Fn::GetAZs": ""
            }
            ],
         "Encrypted": true
         
      }
   }
}
```

**YAML example**

```yaml
EBSVolume:
   Type: AWS::EC2::Volume
   Properties:
      Size: 100
      AvailabilityZone: !Select
         - 0
         - !GetAZs ''
      Encrypted: true
```
CT.EC2.PR.7 rule specification

# # #####################################################################
## Rule Specification ##
# # #####################################################################
#
# Rule Identifier:
#   ec2_encrypted_volumes_check
#
# Description:
#   Checks whether standalone Amazon EC2 EBS volumes and new EC2 EBS volumes created
#   through EC2 instance
#   Block Device Mappings are encrypted at rest.
#
# Reports on:
#   AWS::EC2::Instance, AWS::EC2::Volume
#
# Evaluates:
#   AWS CloudFormation, CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or CloudFormation hook document
#   And: The input document does not contain any Amazon EC2 volume resources
#   Then: SKIP
# Scenario: 2
#   Given: The input document is an AWS CloudFormation or CloudFormation hook document
#   And: The input document contains an EC2 instance resource
#   And: 'BlockDeviceMappings' has not been provided or has been provided as an empty
#   list
#   Then: SKIP
# Scenario: 3
#   Given: The input document is an AWS CloudFormation or CloudFormation hook document
#   And: The input document contains an EC2 instance resource
#   And: 'BlockDeviceMappings' has been provided as a non-empty list
#   And: 'Ebs' has been provided in a 'BlockDeviceMappings' configuration
#   And: 'Encrypted' has not been provided in the 'Ebs' configuration
#   Then: FAIL
# Scenario: 4
#   Given: The input document is an AWS CloudFormation or CloudFormation hook document
#   And: The input document contains an EC2 instance resource
#   And: 'BlockDeviceMappings' has been provided as a non-empty list
#   And: 'Ebs' has been provided in a 'BlockDeviceMappings' configuration
#   And: 'Encrypted' has been provided in the 'Ebs' configuration and set to
#   bool(false)
#   Then: FAIL
# Scenario: 5
#   Given: The input document is an AWS CloudFormation or CloudFormation hook document
#   And: The input document contains an EC2 volume resource
#   And: 'Encrypted' on the EC2 volume has not been provided
#   Then: FAIL
# Scenario: 6
#   Given: The input document is an AWS CloudFormation or CloudFormation hook document
#   And: The input document contains an EC2 volume resource
#   And: 'Encrypted' on the EC2 volume has been provided and is set to bool(false)
#   Then: FAIL
# Scenario: 7
#   Given: The input document is an AWS CloudFormation or CloudFormation hook document
#   And: The input document contains an EC2 instance resource
#   And: 'BlockDeviceMappings' has been provided as a non-empty list
#   And: 'Ebs' has been provided in a 'BlockDeviceMappings' configuration
And: 'Encrypted' has been provided in the 'Ebs' configuration and set to bool(true)
Then: PASS
Scenario: 8
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an EC2 volume resource
And: 'Encrypted' on the EC2 volume has been provided and is set to bool(true)
Then: PASS

# Constants
let EC2_VOLUME_TYPE = "AWS::EC2::Volume"
let EC2_INSTANCE_TYPE = "AWS::EC2::Instance"
let INPUT_DOCUMENT = this

# Assignments
let ec2_volumes = Resources.*[ Type == %EC2_VOLUME_TYPE ]
let ec2_instances = Resources.*[ Type == %EC2_INSTANCE_TYPE ]

# Primary Rules
rule ec2_encrypted_volumes_check when is_cfn_template(%INPUT_DOCUMENT)
%ec2_volumes not empty {
    check_volume(%ec2_volumes.Properties)
    [CT.EC2.PR.7]: Require that an Amazon EBS volume attached to an Amazon EC2 instance is encrypted at rest
    [FIX]: Set 'Encryption' to true on EC2 EBS Volumes.
}
rule ec2_encrypted_volumes_check when is_cfn_hook(%INPUT_DOCUMENT, %EC2_VOLUME_TYPE) {
    check_volume(%INPUT_DOCUMENT.%EC2_VOLUME_TYPE.resourceProperties)
    [CT.EC2.PR.7]: Require that an Amazon EBS volume attached to an Amazon EC2 instance is encrypted at rest
    [FIX]: Set 'Encryption' to true on EC2 EBS Volumes.
}
rule ec2_encrypted_volumes_check when is_cfn_template(%INPUT_DOCUMENT)
%ec2_instances not empty {
    check_instance(%ec2_instances.Properties)
    [CT.EC2.PR.7]: Require that an Amazon EBS volume attached to an Amazon EC2 instance is encrypted at rest
    [FIX]: Set 'Encryption' to true on EC2 EBS Volumes.
}
rule ec2_encrypted_volumes_check when is_cfn_hook(%INPUT_DOCUMENT, %EC2_INSTANCE_TYPE) {
    check_instance(%INPUT_DOCUMENT.%EC2_INSTANCE_TYPE.resourceProperties)
    [CT.EC2.PR.7]: Require that an Amazon EBS volume attached to an Amazon EC2 instance is encrypted at rest
    [FIX]: Set 'Encryption' to true on EC2 EBS Volumes.
}

# Parameterized Rules
rule check_instance(ec2_instance) {
  %ec2_instance[
    filter_ec2_instance_block_device_mappings(this)
    ] {
      BlockDeviceMappings[
        Ebs exists
        Ebs is_struct
      ] {
        check_volume(Ebs)
      }
    }
  }
}

rule check_volume(ec2_volume) {
  %ec2_volume {
    # Scenario 2
    Encrypted exists
    # Scenarios 3 and 4
    Encrypted == true
  }
}

rule filter_ec2_instance_block_device_mappings(ec2_instance) {
  %ec2_instance {
    BlockDeviceMappings exists
    BlockDeviceMappings is_list
    BlockDeviceMappings not empty
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

**CT.EC2.PR.7 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

<table>
<thead>
<tr>
<th>Parameters:</th>
</tr>
</thead>
<tbody>
<tr>
<td>LatestAmiId:</td>
</tr>
<tr>
<td>Description: Region specific latest AMI ID from the Parameter Store</td>
</tr>
<tr>
<td>Type: AWS::SSM::Parameter::Value<a href="">AWS::EC2::Image::Id</a></td>
</tr>
<tr>
<td>Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPC:</td>
</tr>
<tr>
<td>Type: AWS::EC2::VPC</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>CidrBlock: 10.0.0.0/16</td>
</tr>
<tr>
<td>EnableDnsSupport: 'true'</td>
</tr>
</tbody>
</table>
EnableDnsHostnames: 'true'

Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: ''

EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId:
      Ref: LatestAmiId
    InstanceType: t3.micro
    NetworkInterfaces:
      - DeviceIndex: 0
        SubnetId:
          Ref: Subnet
        AssociatePublicIpAddress: false
        BlockDeviceMappings:
          - DeviceName: '/dev/sdm'
            Ebs:
              VolumeType: gp3
              Iops: 200
              Encrypted: true
              DeleteOnTermination: true
              VolumeSize: 20

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Parameters:
  LatestAmiId:
    Description: Region specific latest AMI ID from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'

  Subnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.1.0/24
      AvailabilityZone:
        Fn::Select:
          - 0
          - Fn::GetAZs: ''

  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId:
        Ref: LatestAmiId
      InstanceType: t3.micro
      NetworkInterfaces:
[CT.EC2.PR.8] Require an Amazon EC2 instance to set `AssociatePublicIpAddress` to `false` on a new network interface created by means of the NetworkInterfaces property in the AWS::EC2::Instance resource

This control checks whether your Amazon EC2 instance is configured not to associate a public IP address by default. In particular, this control requires configuring the `AssociatePublicIpAddress` parameter to `false` on a new network interface created by means of the NetworkInterfaces property.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EC2::Instance
- **AWS CloudFormation guard rule:** [CT.EC2.PR.8 rule specification](p. 536)

### Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.EC2.PR.8 rule specification](p. 536)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.EC2.PR.8 example templates](p. 539)

**This control is incompatible with AWS Cloud9**

A compatibility issue exists with AWS Cloud9 and this AWS Control Tower proactive control. If this control is enabled, you cannot create an Amazon EC2 environment in AWS Cloud9. For more information, see [Troubleshooting AWS Cloud9](p. 534).

### Explanation

A public IPv4 address is an IP address that is reachable from the internet. If you launch your instance with a public IP address, then your EC2 instance is reachable from the internet. A private IPv4 address is an IP address that is not reachable from the internet. You can use private IPv4 addresses for communication between EC2 instances in the same VPC or in your connected private network.

IPv6 addresses are globally unique, and therefore are reachable from the internet. However, by default all subnets have the IPv6 addressing attribute set to false.

The network interface settings can be inherited from the Launch Template specified with the LaunchTemplateId property.
Usage considerations

- This control applies only to a new network interface created by means of the `NetworkInterfaces` property, where a `NetworkInterfaceId` has not been specified.
- This control requires subnet information to be specified within a `NetworkInterfaces` configuration instead of the root level `SubnetId` property.
- This control does not check a network interface that may be created in an Amazon EC2 launch template that may be referenced by the `LaunchTemplateName` property.
- A compatibility issue exists with AWS Cloud9 and this AWS Control Tower proactive control. If this control is enabled, you cannot create an Amazon EC2 environment in AWS Cloud9.

Remediation for rule failure

Specify network interfaces using the `NetworkInterfaces` property instead of the root level `SubnetId` property. Set `AssociatePublicIpAddress` to false within each `NetworkInterfaces` configuration.

The examples that follow show how to implement this remediation.

Amazon EC2 Instance - Example

Amazon EC2 instance configured with a new interface that disables public IP address association on creation. The example is shown in JSON and in YAML.

JSON example

```
{
    "EC2Instance": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
            "InstanceType": "t3.micro",
            "ImageId": {
                "Ref": "LatestAmiId"
            },
            "NetworkInterfaces": [
                {
                    "DeviceIndex": 0,
                    "SubnetId": {
                        "Ref": "Subnet"
                    },
                    "AssociatePublicIpAddress": false
                }
            ]
        }
    }
}
```

YAML example

```
EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
        InstanceType: t3.micro
        ImageId: !Ref 'LatestAmiId'
        NetworkInterfaces:
            - DeviceIndex: 0
              SubnetId: !Ref 'Subnet'
```

535
CT.EC2.PR.8 rule specification

# #################################################################
##       Rule Specification        ##
####################################################################
#
# Rule Identifier:
#   ec2_instance_no_public_ip_check
#
# Description:
#   This control checks whether your Amazon EC2 instance is configured to associate a
#   public IP address.
#
# Reports on:
#   AWS::EC2::Instance
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any EC2 instance resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EC2 instance resource
#     And: 'NetworkInterfaces' is not present on the EC2 instance resource or is an empty
#          list
#     And: 'SubnetId' is not provided as a top-level resource property
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EC2 instance resource
#     And: 'NetworkInterfaces' is present on the EC2 instance resource as a non-empty
#          list
#     And: 'NetworkInterfaceId' is present for a configuration in 'NetworkInterfaces' and
#          is a non-empty string or
#          valid local reference
#     Then: SKIP
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EC2 instance resource
#     And: 'NetworkInterfaces' is not provided
#     And: 'SubnetId' is provided as a top-level resource property
#     Then: FAIL
#   Scenario: 5
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EC2 instance resource
#     And: 'NetworkInterfaces' is present on the EC2 instance resource with one or more
#          configurations

AssociatePublicIpAddress: false
And: 'NetworkInterfaceId' is not present or is present and is an empty string or invalid local reference for
   a configuration in 'NetworkInterfaces'
And: 'AssociatePublicIpAddress' is not present for a configuration in 'NetworkInterfaces'
   Then: FAIL
# Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an EC2 instance resource
And: 'NetworkInterfaces' is present on the EC2 instance resource
And: 'NetworkInterfaceId' is not present or is present and is an empty string or invalid local reference for
   a configuration in 'NetworkInterfaces'
And: 'AssociatePublicIpAddress' is present for a configuration in 'NetworkInterfaces'
   And: 'AssociatePublicIpAddress' is set to bool(true)
   Then: FAIL
# Scenario: 7
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an EC2 instance resource
And: 'NetworkInterfaces' is present on the EC2 instance resource
And: 'NetworkInterfaceId' is not present or is present and is an empty string or invalid local reference for
   a configuration in 'NetworkInterfaces'
And: 'AssociatePublicIpAddress' is present for a configuration in 'NetworkInterfaces'
   And: 'AssociatePublicIpAddress' is set to bool(false)
   Then: PASS

# Constants
let EC2_INSTANCE_TYPE = "AWS::EC2::Instance"
let INPUT_DOCUMENT = this

# Assignments
let ec2_instances = Resources.*[ Type == %EC2_INSTANCE_TYPE ]

# Primary Rules
rule ec2_instance_no_public_ip_check when is_cfn_template(%INPUT_DOCUMENT)
   %ec2_instances not empty {
      check(%ec2_instances.Properties)
      <<
      [CT.EC2.PR.8]: Require any Amazon EC2 instance to have a non-public IP address
      [FIX]: Specify network interfaces using the 'NetworkInterfaces' property instead of the root level 'SubnetId' property. Set 'AssociatePublicIpAddress' to false within each 'NetworkInterfaces' configuration.
      >>
   }
rule ec2_instance_no_public_ip_check when is_cfn_hook(%INPUT_DOCUMENT, %EC2_INSTANCE_TYPE)
   {
      check(%INPUT_DOCUMENT.%EC2_INSTANCE_TYPE.resourceProperties)
      <<
      [CT.EC2.PR.8]: Require any Amazon EC2 instance to have a non-public IP address
      [FIX]: Specify network interfaces using the 'NetworkInterfaces' property instead of the root level 'SubnetId' property. Set 'AssociatePublicIpAddress' to false within each 'NetworkInterfaces' configuration.
      >>
   }

537
AWS Control Tower User Guide
Proactive controls

#
# Parameterized Rules
#
rule check(ec2_instance) {
%ec2_instance[ SubnetId exists ] {
# Scenario 5
SubnetId not exists
}

}

%ec2_instance[
# Scenario 2
NetworkInterfaces exists
NetworkInterfaces is_list
NetworkInterfaces not empty
] {
NetworkInterfaces[
# Scenario 3 and 4
filter_network_interfaces(this)
] {
# Scenario 6
AssociatePublicIpAddress exists
# Scenarios 7 and 8
AssociatePublicIpAddress == false
}
}

rule filter_network_interfaces(network_interface) {
%network_interface {
NetworkInterfaceId not exists or
filter_property_is_empty_string(NetworkInterfaceId) or
filter_exclude_valid_local_reference(%INPUT_DOCUMENT, NetworkInterfaceId,
"AWS::EC2::NetworkInterface")
}
}
rule filter_property_is_empty_string(value) {
%value {
this is_string
this == /\A\s*\z/
}
}
rule filter_exclude_valid_local_reference(doc, reference_properties,
referenced_resource_type) {
%reference_properties {
this not is_string
this is_struct
when this.'Fn::GetAtt' exists {
'Fn::GetAtt' {
when filter_query_template_resources(%doc, this[0],
%referenced_resource_type) {
this not exists
}
this exists
}
}
when this.'Fn::GetAtt' not exists {
this exists
}
}

}

rule filter_query_template_resources(doc, resource_key, referenced_resource_type) {

538


let referenced_resource = %doc.Resources[ keys == %resource_key ]
%referenced_resource not empty
%referenced_resource {
  Type in %referenced_resource_type
}
#
# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.EC2.PR.8 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Parameters:
  LatestAmiId:
    Description: Region specific latest AMI ID from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
  Subnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: Ref: VPC
      CidrBlock: 10.0.1.0/24
      AvailabilityZone: Fn::Select:
        - 0
        - Fn::GetAZs: ''
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t3.micro
      ImageId: Ref: LatestAmiId
      NetworkInterfaces:
        - DeviceIndex: 0
          SubnetId:
            Ref: Subnet
          AssociatePublicIpAddress: false
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Parameters:
- LatestAmiId:
  - Description: Region specific latest AMI ID from the Parameter Store
  - Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  - Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
- VPC:
  - Type: AWS::EC2::VPC
  - Properties:
    - CidrBlock: 10.0.0.0/16
    - EnableDnsSupport: 'true'
    - EnableDnsHostnames: 'true'

- Subnet:
  - Type: AWS::EC2::Subnet
  - Properties:
    - VpcId: Ref: VPC
    - CidrBlock: 10.0.1.0/24
    - AvailabilityZone:
      - Fn::Select:
        - 0
        - Fn::GetAZs: ''

- EC2Instance:
  - Type: AWS::EC2::Instance
  - Properties:
    - InstanceType: t3.micro
    - ImageId: Ref: LatestAmiId
    - SubnetId: Ref: Subnet

[CT.EC2.PR.9] Require any Amazon EC2 launch template not to auto-assign public IP addresses to network interfaces

This control checks whether your Amazon EC2 launch templates are configured to assign public IP addresses to network interfaces.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EC2::LaunchTemplate
- **AWS CloudFormation guard rule:** [CT.EC2.PR.9 rule specification (p. 542)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.EC2.PR.9 rule specification (p. 542)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.EC2.PR.9 example templates (p. 545)]

Explanation
A public IP address is an IP address that is reachable from the internet. If you configure your network interfaces with a public IP address, then the resources associated to those network interfaces are reachable from the internet. EC2 resources should not be publicly accessible, because this may allow unintended access to your application servers.

**Usage considerations**

- This control applies only to new network interfaces created by means of the NetworkInterfaceId property in LaunchTemplateData (NetworkInterfaces configurations where a NetworkInterfaceId has not been specified).
- This control requires setting AssociatePublicIpAddress to false on new network interfaces created by means of the NetworkInterfaces property in LaunchTemplateData.

**Remediation for rule failure**

Set AssociatePublicIpAddress to false within each NetworkInterfaces configuration in LaunchTemplateData.

The examples that follow show how to implement this remediation.

**Amazon EC2 Launch Template - Example**

Amazon EC2 launch template configured with a network interface that disables public IP address association. The example is shown in JSON and in YAML.

**JSON example**

```
{
   "EC2LaunchTemplate": {
      "Type": "AWS::EC2::LaunchTemplate",
      "Properties": {
         "LaunchTemplateData": {
            "NetworkInterfaces": [
               {
                  "DeviceIndex": 0,
                  "SubnetId": {
                     "Ref": "Subnet"
                  },
                  "AssociatePublicIpAddress": false
               }
            ]
         }
      }
   }
}
```

**YAML example**

```
EC2LaunchTemplate:
   Type: AWS::EC2::LaunchTemplate
   Properties:
      LaunchTemplateData:
         NetworkInterfaces:
            - DeviceIndex: 0
              SubnetId: !Ref 'Subnet'
              AssociatePublicIpAddress: false
```
CT.EC2.PR.9 rule specification

# #############################################################################
## Rule Specification ##
# #############################################################################

# Rule Identifier:
# ec2_launch_template_public_ip_disabled_check
# Description:
# This control checks whether your Amazon EC2 launch templates are configured to assign public IP addresses to network interfaces.
# Reports on:
# AWS::EC2::LaunchTemplate
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
# None
# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document does not contain any Amazon EC2 launch template resources
#       Then: SKIP
# Scenario: 2
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document contains an Amazon EC2 launch template resource
#       And: 'NetworkInterfaces' is not provided in 'LaunchTemplateData'
#       Then: SKIP
# Scenario: 3
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document contains an Amazon EC2 launch template resource
#       And: 'LaunchTemplateData.NetworkInterfaces' is present on the Amazon EC2 launch template resource as a non empty list
#       And: 'NetworkInterfaceId' is present for a configuration in 'NetworkInterfaces' and is a non-empty string or valid local reference
#       Then: SKIP
# Scenario: 4
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document contains an Amazon EC2 launch template resource
#       And: 'LaunchTemplateData.NetworkInterfaces' is present on the Amazon EC2 launch template resource
#       And: 'NetworkInterfaceId' is not present or is present and is an empty string or invalid local reference for a configuration in 'NetworkInterfaces'
#       And: 'AssociatePublicIpAddress' is not present for a configuration in 'NetworkInterfaces'
#       Then: FAIL
# Scenario: 5
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document contains an Amazon EC2 launch template resource

542
And: 'LaunchTemplateData.NetworkInterfaces' is present on the Amazon EC2 launch template resource
And: 'NetworkInterfaceId' is not present or is present and is an empty string or invalid local reference for
a configuration in 'NetworkInterfaces'
And: 'AssociatePublicIpAddress' is present for a configuration in 'NetworkInterfaces'
And: 'AssociatePublicIpAddress' is set to bool(true)
Then: FAIL
Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon EC2 launch template resource
And: 'LaunchTemplateData.NetworkInterfaces' is present on the Amazon EC2 launch template resource
And: 'NetworkInterfaceId' is not present or is present and is an empty string or invalid local reference for
a configuration in 'NetworkInterfaces'
And: 'AssociatePublicIpAddress' is present for a configuration in 'NetworkInterfaces'
And: 'AssociatePublicIpAddress' is set to bool(false)
Then: PASS

# Constants
let EC2_LAUNCH_TEMPLATE_TYPE = "AWS::EC2::LaunchTemplate"
let INPUT_DOCUMENT = this

# Assignments
let ec2_launch_templates = Resources.*[ Type == %EC2_LAUNCH_TEMPLATE_TYPE ]

# Primary Rules

rule ec2_launch_template_public_ip_disabled_check when is_cfn_template(%INPUT_DOCUMENT)
%ec2_launch_templates not empty {
    check(%ec2_launch_templates.Properties)
    <<<
        [CT.EC2.PR.9]: Require any Amazon EC2 launch template not to auto-assign public IP addresses to network interfaces
        [FIX]: Set 'AssociatePublicIpAddress' to 'false' within each 'NetworkInterfaces' configuration in 'LaunchTemplateData'.
    >>>
}

rule ec2_launch_template_public_ip_disabled_check when is_cfn_hook(%INPUT_DOCUMENT, %EC2_LAUNCH_TEMPLATE_TYPE) {
    check(%INPUT_DOCUMENT.%EC2_LAUNCH_TEMPLATE_TYPE.resourceProperties)
    <<<
        [CT.EC2.PR.9]: Require any Amazon EC2 launch template not to auto-assign public IP addresses to network interfaces
        [FIX]: Set 'AssociatePublicIpAddress' to 'false' within each 'NetworkInterfaces' configuration in 'LaunchTemplateData'.
    >>>
}

# Parameterized Rules

rule check(ec2_launch_templates) {
    %ec2_launch_templates[
        # Scenario 2
        filter_launch_template(this)
AWS Control Tower User Guide
Proactive controls
] {

}

}

LaunchTemplateData {
NetworkInterfaces[
# Scenario 3 and 4
filter_network_interfaces(this)
] {
# Scenario 5 and 6
AssociatePublicIpAddress exists
AssociatePublicIpAddress == false
}
}

rule filter_launch_template(ec2_launch_template) {
%ec2_launch_template {
LaunchTemplateData exists
LaunchTemplateData is_struct
LaunchTemplateData {
NetworkInterfaces exists
NetworkInterfaces is_list
NetworkInterfaces not empty
}
}
}
rule filter_network_interfaces(network_interface) {
%network_interface {
NetworkInterfaceId not exists or
filter_property_is_empty_string(NetworkInterfaceId) or
filter_exclude_valid_local_reference(%INPUT_DOCUMENT, NetworkInterfaceId,
"AWS::EC2::NetworkInterface")
}
}
rule filter_property_is_empty_string(value) {
%value {
this is_string
this == /\A\s*\z/
}
}
rule filter_exclude_valid_local_reference(doc, reference_properties,
referenced_resource_type) {
%reference_properties {
this not is_string
this is_struct

}

}

when this.'Fn::GetAtt' exists {
'Fn::GetAtt' {
when query_for_resource(%doc, this[0], %referenced_resource_type) {
this not exists
}
this exists
}
}
when this.'Fn::GetAtt' not exists {
this exists
}

#
# Utility Rules
#
rule is_cfn_template(doc) {

544


CT.EC2.PR.9 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
  Subnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.1.0/24
      AvailabilityZone:
        Fn::Select:
          - 0
          - Fn::GetAZs: '
  EC2LaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateData:
        NetworkInterfaces:
          - DeviceIndex: 0
            SubnetId:
              Ref: Subnet
            AssociatePublicIpAddress: false

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
EnableDnsSupport: 'true'
EnableDnsHostnames: 'true'

Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: ''

EC2LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateData:
      NetworkInterfaces:
        - DeviceIndex: 0
          SubnetId:
            Ref: Subnet
          AssociatePublicIpAddress: true

[CT.EC2.PR.10] Require Amazon EC2 launch templates to have Amazon CloudWatch detailed monitoring activated

This control checks whether the Amazon EC2 launch template has detailed monitoring enabled.

- Control objective: Establish logging and monitoring
- Implementation: AWS CloudFormation Guard Rule
- Control behavior: Proactive
- Resource types: AWS::EC2::LaunchTemplate
- AWS CloudFormation guard rule: CT.EC2.PR.10 rule specification (p. 547)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.EC2.PR.10 rule specification (p. 547)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.EC2.PR.10 example templates (p. 549)

Explanation

Monitoring is an important part of maintaining the reliability, availability, and performance of your AWS solutions. You should collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. From a security perspective, logging is also an important feature to enable for future forensics efforts in the case of any security incidents.

Remediation for rule failure

In LaunchTemplateData, provide a Monitoring configuration with Enabled set to true.

The examples that follow show how to implement this remediation.

Amazon EC2 Launch Template - Example

Amazon EC2 launch template configured with detailed monitoring enabled. The example is shown in JSON and in YAML.
JSON example

```json
{
    "EC2LaunchTemplate": {
        "Type": "AWS::EC2::LaunchTemplate",
        "Properties": {
            "LaunchTemplateData": {
                "Monitoring": {
                    "Enabled": true
                }
            }
        }
    }
}
```

YAML example

```yaml
EC2LaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
        LaunchTemplateData:
            Monitoring:
                Enabled: true
```

CT.EC2.PR.10 rule specification

```bash
# ###################################
##       Rule Specification        ##
#####################################
#
# Rule Identifier:
#   ec2_launch_template_monitoring_enabled_check
#
# Description:
#   This control checks whether the Amazon EC2 launch template has detailed monitoring
#   enabled.
#
# Reports on:
#   AWS::EC2::LaunchTemplate
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any EC2 launch template resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EC2 launch template resource
```
And: 'LaunchTemplateData.Monitoring.Enabled' has not been provided or has been provided and is empty.
Then: FAIL

Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an EC2 launch template resource
And: 'LaunchTemplateData.Monitoring.Enabled' has been provided
And: 'LaunchTemplateData.Monitoring.Enabled' is equal to a value other than bool(true)
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an EC2 launch template resource
And: 'LaunchTemplateData.Monitoring.Enabled' has been provided
And: 'LaunchTemplateData.Monitoring.Enabled' is equal to bool(true)
Then: PASS

Constants
let EC2_LAUNCH_TEMPLATE_TYPE = "AWS::EC2::LaunchTemplate"
let INPUT_DOCUMENT = this

Assignments
let ec2_launch_templates = Resources.*[ Type == %EC2_LAUNCH_TEMPLATE_TYPE ]

Primary Rules
rule ec2_launch_template_monitoring_enabled_check when is_cfn_template(this)
%ec2_launch_templates not empty {
  check(%ec2_launch_templates.Properties)
  %ec2_launch_templates not empty {
    [CT.EC2.PR.10]: Require Amazon EC2 launch templates to have Amazon CloudWatch
detailed monitoring activated
    [FIX]: In 'LaunchTemplateData', provide a 'Monitoring' configuration with 'Enabled'
set to 'true'.
  }
}

rule ec2_launch_template_monitoring_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %EC2_LAUNCH_TEMPLATE_TYPE) {
  check(%INPUT_DOCUMENT.%EC2_LAUNCH_TEMPLATE_TYPE.resourceProperties)
  %ec2_launch_templates not empty {
    [CT.EC2.PR.10]: Require Amazon EC2 launch templates to have Amazon CloudWatch
detailed monitoring activated
    [FIX]: In 'LaunchTemplateData', provide a 'Monitoring' configuration with 'Enabled'
set to 'true'.
  }
}

Parameterized Rules
rule check(ec2_launch_template) {
  %ec2_launch_template {
    # Scenario 2
    LaunchTemplateData exists
    LaunchTemplateData is_struct
    LaunchTemplateData {
      Monitoring exists
  }
Monitoring is_struct

# Scenario 3 and 4
Monitoring {
  Enabled exists
  Enabled == true
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.EC2.PR.10 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  EC2LaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateData:
        Monitoring:
          Enabled: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  EC2LaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateData:
        Monitoring:
          Enabled: false

[CT.EC2.PR.11] Require that an Amazon EC2 subnet does not automatically assign public IP addresses

This control checks whether your Amazon VPC subnets assign public IP addresses automatically.
• **Control objective:** Limit network access
• **Implementation:** AWS CloudFormation Guard Rule
• **Control behavior:** Proactive
• **Resource types:** AWS::EC2::Subnet
• **AWS CloudFormation guard rule:** [CT.EC2.PR.11 rule specification (p. 552)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.EC2.PR.11 rule specification (p. 552)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.EC2.PR.11 example templates (p. 553)]

**Explanation**

All subnets have an attribute that determines whether a network interface created in the subnet automatically receives a public IPv4 address. When launched into subnets that have this attribute enabled, instances receive a public IP address assigned to their primary network interface.

**Usage considerations**

- This control deactivates automatic assignment of public IP addresses for new network interfaces in Amazon VPC subnets.
- When this control is in operation, public IP addresses can be assigned to network interfaces by means of resource-level settings. (For example, assignment of a public IP address can be made at EC2 instance launch time.)

**Remediation for rule failure**

Omit the `MapPublicIpOnLaunch` property to use the default configuration, or set the `MapPublicIpOnLaunch` property to `false`.

The examples that follow show how to implement this remediation.

**Amazon VPC Subnet - Example One**

Amazon VPC subnet configured to deactivate automatic assignment of public IP addresses by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "Subnet": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "VpcId": {
        "Ref": "VPC"
      },
      "CidrBlock": "10.0.0.0/24",
      "AvailabilityZone": {
        "Fn::Select": [0],
        "Fn::GetAZs": ""
      }
    }
  }
}
```

550
YAML example

Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref 'VPC'
    CidrBlock: 10.0.0.0/24
    AvailabilityZone: !Select
      - 0
      - !GetAZs ''

The examples that follow show how to implement this remediation.

Amazon VPC Subnet - Example Two

Amazon VPC subnet configured to deactivate automatic assignment of public IP addresses by means of the `MapPublicIpOnLaunch` property. The example is shown in JSON and in YAML.

JSON example

```
{
  "Subnet": {
    "Type": "AWS::EC2::Subnet",
    "Properties": {
      "VpcId": {
        "Ref": "VPC"
      },
      "CidrBlock": "10.0.0.0/24",
      "AvailabilityZone": {
        "Fn::Select": [
          0,
          { "Fn:GetAZs": "" }
        ],
        "MapPublicIpOnLaunch": false
      }
    }
  }
}
```

YAML example

```
Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref 'VPC'
    CidrBlock: 10.0.0.0/24
    AvailabilityZone: !Select
      - 0
```

CT.EC2.PR.11 rule specification

```plaintext
# #####################################################################
## Rule Specification       ##
# #####################################################################

# # Rule Identifier:
# subnet_auto_assign_public_ip_disabled_check
# # Description:
# This control checks whether your Amazon VPC subnets automatically assign public IP addresses.
# # Reports on:
# AWS::EC2::Subnet
# # Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
# # Rule Parameters:
# None
# # Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain any EC2 subnet resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EC2 subnet resource
# And: 'MapPublicIpOnLaunch' is present and set to bool(true)
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EC2 subnet resource
# And: 'MapPublicIpOnLaunch' is not present
# Then: PASS
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EC2 subnet resource
# And: 'MapPublicIpOnLaunch' is present and set to bool(false)
# Then: PASS

# Constants
#
let EC2_SUBNET_TYPE = "AWS::EC2::Subnet"
let INPUT_DOCUMENT = this

# Assignments
#
let ec2_subnets = Resources.*[ Type == %EC2_SUBNET_TYPE ]
```
# Primary Rules

rule subnet_auto_assign_public_ip_disabled_check when is_cfn_template(%INPUT_DOCUMENT) {
  %ec2_subnets not empty {
    check(%ec2_subnets.Properties) <<
    [CT.EC2.PR.11]: Require that an Amazon EC2 subnet does not automatically assign public IP addresses
    [FIX]: Omit the 'MapPublicIpOnLaunch' property to use the default configuration, or set the 'MapPublicIpOnLaunch' property to 'false'.
    >>
  }
}

rule subnet_auto_assign_public_ip_disabled_check when is_cfn_hook(%INPUT_DOCUMENT, %EC2_SUBNET_TYPE) {
  check(%INPUT_DOCUMENT.%EC2_SUBNET_TYPE.resourceProperties) <<
  [CT.EC2.PR.11]: Require that an Amazon EC2 subnet does not automatically assign public IP addresses
  [FIX]: Omit the 'MapPublicIpOnLaunch' property to use the default configuration, or set the 'MapPublicIpOnLaunch' property to 'false'.
  >>
}

# Parameterized Rules

rule check(ec2_subnet) {
  %ec2_subnet {
    # Scenario 3
    MapPublicIpOnLaunch not exists or 
    # Scenarios 2 and 4
    MapPublicIpOnLaunch == false
  }
}

# Utility Rules

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.EC2.PR.11 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
  Subnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/24
      AvailabilityZone:
        Fn::Select:
        - 0
        - Fn::GetAZs: ''
      MapPublicIpOnLaunch: true
```

[CT.EC2.PR.12] Require an Amazon EC2 instance to specify at most one network interface by means of the `NetworkInterfaces` property in the `AWS::EC2::Instance` resource

This control checks whether your Amazon Elastic Compute Cloud (Amazon EC2) instance uses multiple ENIs (Elastic Network Interfaces). Specifically, it checks whether an `AWS::EC2::Instance` resource specifies multiple ENIs in the `NetworkInterfaces` property.

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** `AWS::EC2::Instance`
- **AWS CloudFormation guard rule:** [CT.EC2.PR.12 rule specification (p. 556)](https://aws.amazon.com/userguide/aws-control-tower-user-guide/)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.EC2.PR.12 rule specification (p. 556)](https://aws.amazon.com/userguide/aws-control-tower-user-guide/)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.EC2.PR.12 example templates (p. 557)](https://aws.amazon.com/userguide/aws-control-tower-user-guide/)

554
**Explanation**

Multiple ENIs can cause dual-homed instances, meaning instances that have multiple subnets. This duplication can add network security complexity and introduce unintended network paths and access.

The network interface settings can be inherited from the Launch Template specified with the `LaunchTemplateId` property.

**Usage considerations**

- This control does not check a network interface that may be specified in an Amazon EC2 launch template and referenced by the `LaunchTemplateId` property.
- This rule is incompatible with scenarios in which the `NetworkInterfaces` property must be used to specify multiple ENIs. For example, this control may fail if an Amazon EC2 instance that belongs to an Amazon EKS cluster specifies more than one ENI by means of the `NetworkInterfaces` property.

**Remediation for rule failure**

Configure Amazon EC2 instances with only one ENI.

The examples that follow show how to implement this remediation.

**Amazon EC2 Instance - Example**

EC2 Instance with a single network interface. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "EC2Instance": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
            "ImageId": {
                "Ref": "LatestAmiId"
            },
            "NetworkInterfaces": [
                {
                    "SubnetId": {
                        "Ref": "TestSubnet"
                    },
                    "DeviceIndex": 0
                }
            ]
        }
    }
}
```

**YAML example**

```yaml
EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
        ImageId: !Ref 'LatestAmiId'
        NetworkInterfaces:
            - SubnetId: !Ref 'TestSubnet'
              DeviceIndex: 0
```
CT.EC2.PR.12 rule specification

# ####################################################################
##   Rule Specification      ##
# ####################################################################
#
# Rule Identifier:
#   ec2_instance_multiple_eni_check
#
# Description:
#   Checks whether Amazon Elastic Compute Cloud (Amazon EC2) instances use multiple ENIs (Elastic Network Interfaces)
#   or Elastic Fabric Adapters (EFAs).
#
# Reports on:
#   AWS::EC2::Instance
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any Amazon EC2 instance resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document contains an Amazon EC2 instance resource
#     And: 'NetworkInterfaces' is not present or is present and contains 0 configurations
#     Then: SKIP
#   Scenario: 3
#     Given: The input document contains an Amazon EC2 instance resource
#     And: 'NetworkInterfaces' is present and contains >1 configurations
#     Then: FAIL
#   Scenario: 4
#     Given: The input document contains an Amazon EC2 instance resource
#     And: 'NetworkInterfaces' is present
#     And: 'NetworkInterfaces' is present and contains 1 configuration
#     Then: PASS
#
# Constants
#
let EC2_INSTANCE_TYPE = "AWS::EC2::Instance"
let INPUT_DOCUMENT = this
#
# Assignments
#
let ec2_instances = Resources.*[ Type == %EC2_INSTANCE_TYPE ]
#
# Primary Rules
#
rule ec2_instance_multiple_eni_check when is_cfn_template(%INPUT_DOCUMENT)
%ec2_instances not empty {
  check(%ec2_instances.Properties)
  <<
  [CT.EC2.PR.12]: Require an Amazon EC2 instance to configure one ENI only
  [FIX]: Configure Amazon EC2 instances with only one ENI.
  >>
}

rule ec2_instance_multiple_eni_check when is_cfn_hook(%INPUT_DOCUMENT, %EC2_INSTANCE_TYPE) {
  check(%INPUT_DOCUMENT.%EC2_INSTANCE_TYPE.resourceProperties)
  <<
  [CT.EC2.PR.12]: Require an Amazon EC2 instance to configure one ENI only
  [FIX]: Configure Amazon EC2 instances with only one ENI.
  >>
}

# Parameterized Rules
#
rule check(ec2_instance) {

  %ec2_instance [
    # Scenario 2
    NetworkInterfaces exists
    NetworkInterfaces is_list
    NetworkInterfaces not empty
  ] {
    # Scenario 3 and 4
    NetworkInterfaces[1] not exists
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.EC2.PR.12 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Parameters:

LatestAmiId:
  Description: Region specific latest AMI ID from the Parameter Store
  Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
  VPC:
Type: AWS::EC2::VPC
Properties:
  CidrBlock: 10.0.0.0/16
  EnableDnsSupport: 'true'
  EnableDnsHostnames: 'true'
Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: Ref: VPC
    CidrBlock: 10.0.0.0/24
EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: Ref: LatestAmiId
    InstanceType: t3.micro
    NetworkInterfaces:
      - SubnetId: Ref: Subnet
        DeviceIndex: 0
      - SubnetId: Ref: Subnet
        DeviceIndex: 1

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Parameters:
  LatestAmiId:
    Description: Region specific latest AMI ID from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: Ref: VPC
    CidrBlock: 10.0.0.0/24
EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: Ref: LatestAmiId
    InstanceType: t3.micro
    NetworkInterfaces:
      - SubnetId: Ref: Subnet
        DeviceIndex: 0
      - SubnetId: Ref: Subnet
        DeviceIndex: 1
[CT.EC2.PR.13] Require an Amazon EC2 instance to have detailed monitoring enabled

This control checks whether an Amazon EC2 instance has detailed monitoring enabled.

- **Control objective**: Establish logging and monitoring
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::EC2::Instance
- **AWS CloudFormation guard rule**: CT.EC2.PR.13 rule specification (p. 561)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: CT.EC2.PR.13 rule specification (p. 561)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.EC2.PR.13 example templates (p. 562)

**Explanation**

By default, all Amazon EC2 instances are created with basic monitoring that sends host-level logs to Amazon CloudWatch every five (5) minutes. With detailed monitoring, host-level logs are collected every one (1) minute instead, leading to faster detection of possible malicious or anomalous activity.

**Usage considerations**

- When you enable detailed monitoring, you are charged per metric that is sent to CloudWatch. You are not charged for data storage. For more information, see the Amazon CloudWatch pricing page.

**Remediation for rule failure**

Set Monitoring to true.

The examples that follow show how to implement this remediation.

**EC2 Instance - Example**

An EC2 Instance with detailed monitoring enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "Parameters": {
        "LatestAmiId": {
            "Description": "Region specific latest AMI ID from the Parameter Store",
            "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
            "Default": "/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2"
        }
    },
    "Resources": {
        "VPC": {
            "Type": "AWS::EC2::VPC",
            "Properties": {
                "CidrBlock": "10.0.0.0/16",
                "EnableDnsSupport": "true",
                "EnableDnsHostnames": "true"
            }
        }
    }
}
```
YAML example

Parameters:
- LatestAmiId:
  - Description: Region specific latest AMI ID from the Parameter Store
  - Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  - Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
- VPC:
  - Type: AWS::EC2::VPC
  - Properties:
    - CidrBlock: 10.0.0.0/16
    - EnableDnsSupport: 'true'
    - EnableDnsHostnames: 'true'
- Subnet:
  - Type: AWS::EC2::Subnet
  - Properties:
    - VpcId: !Ref 'VPC'
    - CidrBlock: 10.0.0.0/24
- EC2Instance:
  - Type: AWS::EC2::Instance
  - Properties:
    - ImageId: !Ref 'LatestAmiId'
    - InstanceType: t3.micro
    - NetworkInterfaces:
      - SubnetId: !Ref 'Subnet'
      - DeviceIndex: 0
  - Monitoring: true
# Rule Specification

## Rule Identifier:
- ec2_instance_detailed_monitoring_enabled_check

## Description:
This control checks whether an Amazon EC2 instance has detailed monitoring enabled.

## Reports on:
- AWS::EC2::Instance

## Evaluates:
- AWS CloudFormation, AWS CloudFormation hook

## Rule Parameters:
- None

## Scenarios:

### Scenario: 1
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document does not contain any EC2 instance resources
- Then: SKIP

### Scenario: 2
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document contains an EC2 instance resource
- And: 'Monitoring' has not been provided
- Then: FAIL

### Scenario: 3
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document contains an EC2 instance resource
- And: 'Monitoring' has been provided and set to a value other than bool(true)
- Then: FAIL

### Scenario: 4
- Given: The input document is an AWS CloudFormation or AWS CloudFormation Hook Document
- And: The input document contains an EC2 instance resource
- And: 'Monitoring' has been provided and set to bool(true)
- Then: PASS

## Constants

```plaintext
let EC2_INSTANCE_TYPE = "AWS::EC2::Instance"
let INPUT_DOCUMENT = this
```

## Assignments

```plaintext
let ec2_instances = Resources.*[ Type == %EC2_INSTANCE_TYPE ]
```

## Primary Rules

```plaintext
rule ec2_instance_detailed_monitoring_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
  %ec2_instances not empty {
    check(%ec2_instances.Properties)
  }
  <<
```
CT.EC2.PR.13: Require an Amazon EC2 instance to have detailed monitoring enabled

[Fix]: Set 'Monitoring' to 'true'.

```
rule ec2_instance_detailed_monitoring_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %EC2_INSTANCE_TYPE) {
    check(%INPUT_DOCUMENT.%EC2_INSTANCE_TYPE.resourceProperties)
    <<
    [CT.EC2.PR.13]: Require an Amazon EC2 instance to have detailed monitoring enabled
    [Fix]: Set 'Monitoring' to 'true'.
    >>
}
```

### Parameterized Rules

```
# Parameterized Rules
#
rule check(ec2_instance) { 
    %ec2_instance {
        # Scenario 2
        Monitoring exists
        # Scenarios 3 and 4
        Monitoring == true
    }
}
```

### Utility Rules

```
# Utility Rules
#
rule is_cfn_template(doc) { 
    %doc {
        AWSTemplateFormatVersion exists  or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) { 
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

### CT.EC2.PR.13 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

**Parameters:**
- **LatestAmiId**:
  - Description: Region specific latest AMI ID from the Parameter Store
  - Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  - Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

**Resources:**
- **VPC**:
  - Type: AWS::EC2::VPC
  - Properties:
    - CidrBlock: 10.0.0.0/16
    - EnableDnsSupport: 'true'
    - EnableDnsHostnames: 'true'

- **Subnet**:
  - Type: AWS::EC2::Subnet
Properties:
  VpcId:
    Ref: VPC
  CidrBlock: 10.0.0.0/24

EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId:
      Ref: LatestAmiId
    InstanceType: t3.micro
    NetworkInterfaces:
      - SubnetId:
          Ref: Subnet
        DeviceIndex: 0
        Monitoring: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Parameters:
  LatestAmiId:
    Description: Region specific latest AMI ID from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
  Subnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/24
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId:
        Ref: LatestAmiId
      InstanceType: t3.micro
      NetworkInterfaces:
        - SubnetId:
            Ref: Subnet
          DeviceIndex: 0
          Monitoring: false

[CT.EC2.PR.14] Require an Amazon EBS volume configured through an Amazon EC2 launch template to encrypt data at rest

This control checks whether an Amazon EC2 launch template with EBS volume block device mappings is configured to enable EBS volume encryption.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
• **Resource types**: AWS::EC2::LaunchTemplate

• **AWS CloudFormation guard rule**: [CT.EC2.PR.14 rule specification (p. 565)](#)

### Details and examples

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: [CT.EC2.PR.14 rule specification (p. 565)](#)

• For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.EC2.PR.14 example templates (p. 567)](#)

### Explanation

For an added layer of security of your sensitive data in an EBS volume, you should enable EBS encryption at rest. Amazon EBS encryption offers a straightforward encryption solution for your EBS resources. It doesn't require you to build, maintain, and secure your own key management infrastructure, and it uses KMS keys when creating encrypted volumes and snapshots.

**Usage considerations**

• This control applies only to an EC2 launch template that specifies EBS block device mappings.

• When you launch an instance using a launch template, you can override parameters that are specified in the launch template. To ensure that encryption is enabled for EBS block device mappings when you launch an instance with a launch template by means of the AWS::EC2::Instance resource, use this control in conjunction with CT.EC2.PR.7.

### Remediation for rule failure

For every entry in the `BlockDeviceMappings` parameter with an `Ebs` configuration, set `Encryption` to true.

The examples that follow show how to implement this remediation.

**Amazon EC2 launch template - Example**

An Amazon EC2 launch template configured with an EBS block device mapping that has volume encryption enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "LaunchTemplate": {
    "Type": "AWS::EC2::LaunchTemplate",
    "Properties": {
      "LaunchTemplateData": {
        "BlockDeviceMappings": [
          {
            "DeviceName": "/dev/sdc",
            "Ebs": {
              "Encrypted": true
            }
          }
        ]
      }
    }
  }
}
```
YAML example

LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
Properties:
  LaunchTemplateData:
    BlockDeviceMappings:
      - DeviceName: /dev/sdc
        Ebs:
          Encrypted: true

CT.EC2.PR.14 rule specification

# ###################################
##       Rule Specification        ##
####################################
#
# Rule Identifier:
#   ec2_launch_template_encrypted_volumes_check
#
# Description:
#   This control checks whether an Amazon EC2 launch template with EBS volume block device
#   mappings is configured to enable EBS volume encryption.
#
# Reports on:
#   AWS::EC2::LaunchTemplate
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#     And: The input document does not contain any EC2 launch template resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#     And: The input document contains an EC2 launch template resource
#     And: 'BlockDeviceMappings' in 'LaunchTemplateData' has not been provided or has
#          been provided as an empty list
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#     And: The input document contains an EC2 launch template resource
#     And: 'BlockDeviceMappings' in 'LaunchTemplateData' been provided as a non-empty
#          list
#     And: No entries in 'BlockDeviceMappings' contain 'Ebs' as a struct
#     Then: SKIP
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#     And: The input document contains an EC2 launch template resource
#     And: 'BlockDeviceMappings' in 'LaunchTemplateData' been provided as a non-empty
#          list
#     And: No entries in 'BlockDeviceMappings' contain 'Ebs' as a struct
#     Then: SKIP
# And: An entry in 'BlockDeviceMappings' contains 'Ebs' as a struct
# And: In the same entry, 'Encrypted' in 'Ebs' has not been provided or has been
#       provided and set to a value other than bool(true)
# Then: FAIL

# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#       And: The input document contains an EC2 launch template resource
#       And: 'BlockDeviceMappings' in 'LaunchTemplateData' been provided as a non-empty
#       list
#       And: An entry in 'BlockDeviceMappings' contains 'Ebs' as a struct
#       And: In the same entry, 'Encrypted' in 'Ebs' has not been provided or has been
#       provided and set to bool(true)
# Then: PASS

# Constants
let EC2_LAUNCH_TEMPLATE_TYPE = "AWS::EC2::LaunchTemplate"
let INPUT_DOCUMENT = this

# Assignments
let ec2_launch_templates = Resources.*[ Type == %EC2_LAUNCH_TEMPLATE_TYPE ]

# Primary Rules
# rule ec2_launch_template_encrypted_volumes_check when is_cfn_template(this) {%
%ec2_launch_templates not empty {
    check(%ec2_launch_templates.Properties)
        %ec2_launch_templates not empty {
            [CT.EC2.PR.14]: Require an Amazon EBS volume configured through an Amazon EC2
            launch template to encrypt data at rest
            [FIX]: For every entry in the 'BlockDeviceMappings' parameter with an 'Ebs'
            configuration, set 'Encryption' to true.
        }
    }
# rule ec2_launch_template_encrypted_volumes_check when is_cfn_hook(%INPUT_DOCUMENT,
%EC2_LAUNCH_TEMPLATE_TYPE) {%
    check(%INPUT_DOCUMENT.%EC2_LAUNCH_TEMPLATE_TYPE.resourceProperties)
        %INPUT_DOCUMENT.%EC2_LAUNCH_TEMPLATE_TYPE.resourceProperties {
            [CT.EC2.PR.14]: Require an Amazon EBS volume configured through an Amazon EC2
            launch template to encrypt data at rest
            [FIX]: For every entry in the 'BlockDeviceMappings' parameter with an 'Ebs'
            configuration, set 'Encryption' to true.
        }
    }
# Parameterized Rules
# rule check(ec2_launch_template) {%
%ec2_launch_template [ %ec2_launch_template [ 
    # Scenarios 2 and 3
    filter_launch_template_contains_ebs_block_device_mappings(this)
    ] { LaunchTemplateData { 
        BlockDeviceMappings[
            Ebs exists
            Ebs is_struct
        ]
    ]} {
Ebs {
    # Scenarios 4 and 5
    Encrypted exists
    Encrypted == true
}
}
}
}
}
}

rule filter_launch_template_contains_ebs_block_device_mappings(launch_template) {
    %launch_template {
        LaunchTemplateData exists
        LaunchTemplateData is_struct

        LaunchTemplateData {
            BlockDeviceMappings exists
            BlockDeviceMappings is_list
            BlockDeviceMappings not empty

            some BlockDeviceMappings[*] {
                Ebs exists
                Ebs is_struct
            }
        }
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.EC2.PR.14 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
LaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
        LaunchTemplateData:
            BlockDeviceMappings:
                - DeviceName: /dev/sdc
                  Ebs:
                      Encrypted: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.
Resources:
LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
Properties:
  LaunchTemplateData:
    BlockDeviceMappings:
      - DeviceName: /dev/sdc
    Ebs:
      Encrypted: false

Amazon Elastic Compute Cloud (Amazon EC2) Auto Scaling controls

Topics
  - [CT.AUTOSCALING.PR.1] Require an Amazon EC2 Auto Scaling group to have multiple Availability Zones (p. 568)
  - [CT.AUTOSCALING.PR.2] Require an Amazon EC2 Auto Scaling group launch configuration to configure Amazon EC2 instances for IMDSv2 (p. 572)
  - [CT.AUTOSCALING.PR.3] Require an Amazon EC2 Auto Scaling launch configuration to have a single-hop metadata response limit (p. 576)
  - [CT.AUTOSCALING.PR.4] Require an Amazon EC2 Auto Scaling group associated with an AWS Elastic Load Balancing (ELB) to have ELB health checks activated (p. 581)
  - [CT.AUTOSCALING.PR.5] Require than an Amazon EC2 Auto Scaling group launch configuration does not have Amazon EC2 instances with public IP addresses (p. 585)
  - [CT.AUTOSCALING.PR.6] Require any Amazon EC2 Auto Scaling groups to use multiple instance types (p. 589)
  - [CT.AUTOSCALING.PR.8] Require an Amazon EC2 Auto Scaling group to have EC2 launch templates configured (p. 597)

[CT.AUTOSCALING.PR.1] Require an Amazon EC2 Auto Scaling group to have multiple Availability Zones

This control checks whether your Amazon EC2 Auto Scaling group spans multiple Availability Zones.

  - Control objective: Improve availability
  - Implementation: AWS CloudFormation Guard Rule
  - Control behavior: Proactive
  - Resource types: AWS::AutoScaling::AutoScalingGroup
  - AWS CloudFormation guard rule: CT.AUTOSCALING.PR.1 rule specification (p. 570)

Details and examples
  - For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.AUTOSCALING.PR.1 rule specification (p. 570)
  - For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.AUTOSCALING.PR.1 example templates (p. 571)
Amazon EC2 Auto Scaling groups can be configured to use multiple Availability Zones. An Auto Scaling group with a single Availability Zone is preferred in some use cases, such as batch-jobs or when inter-AZ transfer costs need to be kept to a minimum. However, an Auto Scaling group that does not span multiple Availability Zones will not launch instances in another Availability Zone to compensate if the configured single Availability Zone becomes unavailable.

**Remediation for rule failure**

Configure Auto Scaling groups with multiple Availability Zones.

The examples that follow show how to implement this remediation.

**Auto Scaling group - Example**

Auto Scaling group configured with multiple Availability Zones. The example is shown in JSON and in YAML.

**JSON example**

```
{
    "AutoScalingGroup": {
        "Type": "AWS::AutoScaling::AutoScalingGroup",
        "Properties": {
            "LaunchTemplate": {
                "LaunchTemplateId": {
                    "Ref": "LaunchTemplate"
                },
                "Version": {
                    "Fn::GetAtt": "LaunchTemplate.LatestVersionNumber"
                }
            },
            "MaxSize": "1",
            "MinSize": "0",
            "DesiredCapacity": "1",
            "AvailabilityZones": [
                {
                    "Fn::Select": [
                        0,
                        {
                            "Fn::GetAZs": ""
                        }
                    ],
                },
                {
                    "Fn::Select": [
                        1,
                        {
                            "Fn::GetAZs": ""
                        }
                    ]
                }
            ]
        }
    }
}
```

**YAML example**

```
AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
```
LaunchTemplate:
  LaunchTemplateId: !Ref 'LaunchTemplate'
  Version: !GetAtt 'LaunchTemplate.LatestVersionNumber'
  MaxSize: '1'
  MinSize: '0'
  DesiredCapacity: '1'
  AvailabilityZones:
    - !Select
    - 0
    - !GetAZs ''
    - !Select
    - 1
    - !GetAZs ''

CT.AUTOSCALING.PR.1 rule specification

# CT.AUTOSCALING.PR.1 rule specification

Rule Identifier:
  autoscaling_multiple_az_check

Description:
  Checks if Auto Scaling groups span multiple Availability Zones.

Reports on:
  AWS::AutoScaling::AutoScalingGroup

Evaluates:
  AWS CloudFormation, AWS CloudFormation hook

Rule Parameters:
  None

Scenarios:
  Scenario: 1
  Given: The input document is an AWS CloudFormation or CloudFormation hook document
  And: The input document does not contain any Auto Scaling groups
  Then: SKIP

  Scenario: 2
  Given: The input document is an AWS CloudFormation or CloudFormation hook document
  And: The input document contains an Auto Scaling group resource
  And: 'AvailabilityZones' is not present on the Auto Scaling group resource
  Then: FAIL

  Scenario: 3
  Given: The input document is an AWS CloudFormation or CloudFormation hook document
  And: The input document contains an Auto Scaling group resource
  And: 'AvailabilityZones' is present on the Auto Scaling group resource
  And: The number of 'AvailabilityZones' present is less than 2 (< 2) or the number of unique 'AvailabilityZones' provided is less than 2 (< 2)
  Then: FAIL

  Scenario: 4
  Given: The input document is an AWS CloudFormation or CloudFormation Hook Document
  And: The input document contains an Auto Scaling group resource
  And: 'AvailabilityZones' is present on the Auto Scaling group resource
  And: The number of 'AvailabilityZones' present is greater than or equal to 2 (>= 2)
  And: At least two unique 'AvailabilityZones' have been provided
  Then: PASS
CT.AUTOSCALING.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Parameters:
  LatestAmiId:  
  Description: Region specific latest AMI ID from the Parameter Store  
  Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>  
  Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
  LaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateData:
        ImageId:
          Ref: LatestAmiId
        InstanceType: t3.micro
  AutoScalingGroup:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      LaunchTemplate:
        LaunchTemplateId:
          Ref: LaunchTemplate
        Version:
          Fn::GetAtt: LaunchTemplate.LatestVersionNumber
      MaxSize: '1'
      MinSize: '0'
      DesiredCapacity: '1'
      AvailabilityZones:
        - Fn::Select:
        - 0
        - Fn::GetAZs: ""
        - Fn::Select:
        - 1
        - Fn::GetAZs: ""

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Parameters:
  LatestAmiId:  
  Description: Region specific latest AMI ID from the Parameter Store  
  Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>  
  Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
  LaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateData:
        ImageId:
          Ref: LatestAmiId
        InstanceType: t3.micro
  AutoScalingGroup:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      LaunchTemplate:
        LaunchTemplateId:
          Ref: LaunchTemplate
        Version:
**[CT.AUTOCLUSTER.SHIFT.2]** Require an Amazon EC2 Auto Scaling group launch configuration to configure Amazon EC2 instances for IMDSv2

This control checks whether an Amazon EC2 Auto Scaling launch configuration is configured to require the use of Instance Metadata Service Version 2 (IMDSv2).

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::AutoScaling::LaunchConfiguration

**AWS CloudFormation guard rule:** [CT.AUTOCLUSTER.SHIFT.2 rule specification](p. 573)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.AUTOCLUSTER.SHIFT.2 rule specification](p. 573)
- For examples of PASS and FAIL CloudFront Templates related to this control, see: [CT.AUTOCLUSTER.SHIFT.2 example templates](p. 575)

**Explanation**

IMDS provides data about your instance, which you can use to configure or manage the running instance. Version 2 of the IMDS adds protections that weren't available in IMDSv1, to safeguard your EC2 instances further.

**Usage considerations**

- This control applies only to Amazon EC2 Auto Scaling launch configurations that allow access to instance metadata.

**Remediation for rule failure**

Provide a MetadataOptions configuration and set the value of HttpTokens to required.

The examples that follow show how to implement this remediation.

**Amazon EC2 Auto Scaling Launch Configuration - Example**

Amazon EC2 Auto Scaling launch configuration with IMDSv2 enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "Fn::GetAtt": "LaunchTemplate.LatestVersionNumber",
  "MaxSize": "1",
  "MinSize": "0",
  "DesiredCapacity": "1",
  "AvailabilityZones":
    ["Fn::Select":
      0,
      "Fn::GetAZs": ""
    ]
}
```
"AutoScalingLaunchConfiguration": {
    "Type": "AWS::AutoScaling::LaunchConfiguration",
    "Properties": {
        "ImageId": {
            "Ref": "LatestAmiId"
        },
        "InstanceType": "t3.micro",
        "MetadataOptions": {
            "HttpTokens": "required"
        }
    }
}

YAML example

AutoScalingLaunchConfiguration:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
    ImageId: !Ref 'LatestAmiId'
    InstanceType: t3.micro
    MetadataOptions: 
      HttpTokens: required

CT.AUTOSCALING.PR.2 rule specification

# ###################################################################
#        Rule Specification          #
# ###################################################################
#
# Rule Identifier:
#  autoscaling_launch_config_requires_imdsv2_check
#
# Description:
#  This control checks whether an Amazon EC2 Auto Scaling launch configuration is
#  configured to require the use of Instance Metadata Service Version 2 (IMDSv2).
#
# Reports on:
#  AWS::AutoScaling::LaunchConfiguration
#
# Evaluates:
#  AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#  None
#
# Scenarios:
#  Scenario: 1
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#           And: The input document does not contain any Autoscaling launch configuration
#           resources
#           Then: SKIP
#  Scenario: 2
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#           And: The input document contains an Autoscaling launch configuration resource
#           And: 'MetadataOptions' has been provided.
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Autoscaling launch configuration resource
# And: 'MetadataOptions.HttpEndpoint' has not been provided or has been provided and is equal to 'enabled'
# And: 'MetadataOptions.HttpTokens' has not been provided
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Autoscaling launch configuration resource
# And: 'MetadataOptions' has been provided.
# And: 'MetadataOptions.HttpEndpoint' has not been provided or has been provided and is equal to 'enabled'
# And: 'MetadataOptions.HttpTokens' has been provided and set to a value other than 'required'
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Autoscaling launch configuration resource
# And: 'MetadataOptions' has been provided.
# And: 'MetadataOptions.HttpEndpoint' has not been provided or has been provided and is equal to 'enabled'
# And: 'MetadataOptions.HttpTokens' has been provided and set to 'required'
# Then: PASS

# Constants
# let AUTOSCALING_LAUNCH_CONFIGURATION_TYPE = "AWS::AutoScaling::LaunchConfiguration"
# let INPUT_DOCUMENT = this

# Assignments
# let autoscaling_launch_configurations = Resources.[ Type == %AUTOSCALING_LAUNCH_CONFIGURATION_TYPE ]

# Primary Rules
# rule autoscaling_launch_configRequires IMDSv2 check when is_cfn_template(%INPUT_DOCUMENT)

%autoscaling_launch_configurations not empty {
  check(%autoscaling_launch_configurations.Properties)
  <<
    [CT.AUTOSCALING.PR.2]: Require an Amazon EC2 Auto Scaling group launch configuration to configure Amazon EC2 instances for IMDSv2
    [FIX]: Provide a 'MetadataOptions' configuration and set the value of 'HttpTokens' to 'required'.
  >>
}

rule autoscaling_launch_configRequires IMDSv2 check when is_cfn_hook(%INPUT_DOCUMENT, %AUTOSCALING_LAUNCH_CONFIGURATION_TYPE) {
  check(%INPUT_DOCUMENT.%AUTOSCALING_LAUNCH_CONFIGURATION_TYPE.resourceProperties)
  <<
    [CT.AUTOSCALING.PR.2]: Require an Amazon EC2 Auto Scaling group launch configuration to configure Amazon EC2 instances for IMDSv2
    [FIX]: Provide a 'MetadataOptions' configuration and set the value of 'HttpTokens' to 'required'.
  >>
CT.AUTOSCALING.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Parameters:
LatestAmiId:
   Description: Region specific latest AMI ID from the Parameter Store
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Parameters:
- LatestAmiId:
  - Description: Region specific latest AMI ID from the Parameter Store
  - Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  - Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
AutoScalingLaunchConfiguration:
- Type: AWS::AutoScaling::LaunchConfiguration
- Properties:
  - ImageId:
    - Ref: LatestAmiId
  - InstanceType: t3.micro
  - MetadataOptions:
    - HttpTokens: optional

[CT.AUTOSCALING.PR.3] Require an Amazon EC2 Auto Scaling launch configuration to have a single-hop metadata response limit

This control checks whether an Amazon EC2 Auto Scaling launch configuration has a metadata token hop limit set to 1.

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::AutoScaling::LaunchConfiguration
- **AWS CloudFormation guard rule:** [CT.AUTOSCALING.PR.3 rule specification](p. 578)

Details and examples
- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.AUTOSCALING.PR.3 rule specification](p. 578)
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.AUTOSCALING.PR.3 example templates](p. 581)

Explanation

The Instance Metadata Service (IMDS) provides metadata information about an EC2 instance, which is useful for application configuration. Restricting the HTTP PUT response for the metadata service to the EC2 instance protects the IMDS from unauthorized use.
The Time To Live (TTL) field in the IP packet is reduced by one on every hop. This reduction can be used to ensure that the packet does not travel outside EC2. IMDSv2 protects EC2 instances that may have been misconfigured as open routers, layer 3 firewalls, VPNs, tunnels, or NAT devices, which prevents unauthorized users from retrieving metadata. With IMDSv2, the PUT response that contains the secret token cannot travel outside the instance, because the default metadata response hop limit is set to 1. However, if this value is greater than 1, the token can leave the EC2 instance.

**Usage considerations**

- This control applies only to Amazon EC2 Auto Scaling launch configurations that allow access to instance metadata.
- This control is incompatible with Amazon EC2 Auto Scaling launch configurations that require a token hop limit of 2.

**Remediation for rule failure**

Provide a MetadataOptions configuration with HttpPutResponseLimit set to 1.

The examples that follow show how to implement this remediation.

**Amazon EC2 Auto Scaling Launch Configuration - Example One**

Amazon EC2 Auto Scaling launch configuration configured with access to instance metadata enabled by means of AWS CloudFormation defaults with a token hop limit of 1. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "AutoScalingLaunchConfiguration": {
      "Type": "AWS::AutoScaling::LaunchConfiguration",
      "Properties": {
         "ImageId": {
            "Ref": "LatestAmiId"
         },
         "InstanceType": "t3.micro",
         "MetadataOptions": {
            "HttpPutResponseHopLimit": 1
         }
      }
   }
}
```

**YAML example**

```yaml
AutoScalingLaunchConfiguration:
   Type: AWS::AutoScaling::LaunchConfiguration
   Properties:
      ImageId: !Ref 'LatestAmiId'
      InstanceType: t3.micro
      MetadataOptions:
         HttpPutResponseHopLimit: 1
```

The examples that follow show how to implement this remediation.
Amazon EC2 Auto Scaling Launch Configuration - Example Two

Amazon EC2 Auto Scaling launch configuration configured with access to instance metadata enabled by means of the MetadataOptions property with a token hop limit of 1. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "AutoScalingLaunchConfiguration": {
    "Type": "AWS::AutoScaling::LaunchConfiguration",
    "Properties": {
      "ImageId": {
        "Ref": "LatestAmiId"
      },
      "InstanceType": "t3.micro",
      "MetadataOptions": {
        "HttpEndpoint": "enabled",
        "HttpPutResponseHopLimit": 1
      }
    }
  }
}
```

**YAML example**

```yaml
AutoScalingLaunchConfiguration:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
    ImageId: !Ref 'LatestAmiId'
    InstanceType: t3.micro
    MetadataOptions:
      HttpEndpoint: enabled
      HttpPutResponseHopLimit: 1
```

**CT.AUTOSCALING.PR.3 rule specification**

```bash
#-rule
#-specification
#
#-rule:
#  identifier: autoscaling_launch_config_hop_limit_check
#-description:
#  This control checks whether an Amazon EC2 Auto Scaling launch configuration has a
#  metadata token hop limit set to '1'.
#-reports-on:
#  AWS::AutoScaling::LaunchConfiguration
#-evaluates:
#  AWS CloudFormation, AWS CloudFormation hook
#-rule-parameters:
#  None
```
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document does not contain any Amazon EC2 Auto Scaling launch
configuration resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Amazon EC2 Auto Scaling launch configuration
resource
# And: 'MetadataOptions' has been provided.
# And: 'MetadataOptions.HttpEndpoint' has been provided is equal to 'disabled'
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Amazon EC2 Auto Scaling launch configuration
resource
# And: 'MetadataOptions' has not been provided.
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Amazon EC2 Auto Scaling launch configuration
resource
# And: 'MetadataOptions' has been provided.
# And: 'MetadataOptions.HttpEndpoint' has not been provided or has been provided and
is equal to 'enabled'
# And: 'MetadataOptions.HttpPutResponseHopLimit' has not been provided.
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Amazon EC2 Auto Scaling launch configuration
resource
# And: 'MetadataOptions' has been provided.
# And: 'MetadataOptions.HttpEndpoint' has not been provided or has been provided and
is equal to 'enabled'
# And: 'MetadataOptions.HttpPutResponseHopLimit' has been provided but is not equal
to an integer of '1'.
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Amazon EC2 Auto Scaling launch configuration
resource
# And: 'MetadataOptions' has been provided.
# And: 'MetadataOptions.HttpEndpoint' has not been provided or has been provided and
is equal to 'enabled'
# And: 'MetadataOptions.HttpPutResponseHopLimit' has been provided and is equal to
an integer of '1'.
# Then: PASS

# Constants
let AUTOSCALING_LAUNCH_CONFIG_TYPE = "AWS::AutoScaling::LaunchConfiguration"
let INPUT_DOCUMENT = this

# Assignments
let autoscaling_launch_configurations = Resources.[ Type ==
%AUTOSCALING_LAUNCH_CONFIG_TYPE]
# Primary Rules

## Proactive controls

```plaintext
# Primary Rules

## Proactive controls

```
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.RESOURCE_TYPE.resourceProperties exists
}

CT.AUTOSCALING.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Parameters:
  LatestAmiId:
    Description: Region specific latest AMI ID from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
  AutoScalingLaunchConfiguration:
    Type: AWS::AutoScaling::LaunchConfiguration
    Properties:
      ImageId:
        Ref: LatestAmiId
      InstanceType: t3.micro

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Parameters:
  LatestAmiId:
    Description: Region specific latest AMI ID from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
  AutoScalingLaunchConfiguration:
    Type: AWS::AutoScaling::LaunchConfiguration
    Properties:
      ImageId:
        Ref: LatestAmiId
      InstanceType: t3.micro
      MetadataOptions:
        HttpEndpoint: enabled
        HttpPutResponseHopLimit: 2

[CT.AUTOSCALING.PR.4] Require an Amazon EC2 Auto Scaling group associated with an AWS Elastic Load Balancing (ELB) to have ELB health checks activated

This control checks whether your Amazon EC2 Auto Scaling groups that are associated with a load balancer are using Elastic Load Balancing health checks.
• **Control objective:** Improve availability
• **Implementation:** AWS CloudFormation Guard Rule
• **Control behavior:** Proactive
• **Resource types:** AWS::AutoScaling::AutoScalingGroup
• **AWS CloudFormation guard rule:** [CT.AUTOSCALING.PR.4 rule specification](p. 584)

Details and examples

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.AUTOSCALING.PR.4 rule specification](p. 584)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: [GitHub](129x619)

Explanation

This configuration requirement ensures that the group can determine an instance's health based on additional tests provided by the load balancer. Using Elastic Load Balancing health checks can help support the availability of applications that use EC2 Auto Scaling groups.

**Usage considerations**

• This control only applies to Auto Scaling groups associated with a Classic Load Balancer or Target Group

Remediation for rule failure

Configure Amazon EC2 Auto Scaling groups associated with an Elastic Load Balancing to use Elastic Load Balancing health checks.

The examples that follow show how to implement this remediation.

**Auto Scaling group - Example One**

Auto Scaling group with a Classic Load Balancer association and Elastic Load Balancing health checks. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "AutoScalingGroup": {
        "Type": "AWS::AutoScaling::AutoScalingGroup",
        "Properties": {
            "LaunchTemplate": {
                "LaunchTemplateId": {
                    "Ref": "LaunchTemplate"
                },
                "Version": {
                    "Fn::GetAtt": "LaunchTemplate.LatestVersionNumber"
                }
            },
            "MaxSize": "1",
            "MinSize": "0",
            "DesiredCapacity": "1",
            "LoadBalancerNames": [
                { "Ref": "ElasticLoadBalancer" }
            ],
        }
    }
}
```
The examples that follow show how to implement this remediation.

Auto Scaling group - Example Two

Auto Scaling group with a Target Group association and Elastic Load Balancing health checks. The example is shown in JSON and in YAML.

JSON example

```json
{
   "AutoScalingGroup": {
      "Type": "AWS::AutoScaling::AutoScalingGroup",
      "Properties": {
         "LaunchTemplate": {
            "LaunchTemplateId": {
               "Ref": "LaunchTemplate"
            },
            "Version": {
               "Fn::GetAtt": "LaunchTemplate.LatestVersionNumber"
            }
         },
         "MaxSize": "1",
         "MinSize": "0",
         "DesiredCapacity": "1",
         "TargetGroupARNs": [
            {
               "Ref": "ELBv2TargetGroup"
            }
         ],
         "HealthCheckType": "ELB",
         "VPCZoneIdentifier": [
```

YAML example

```
AutoScalingGroup:
   Type: AWS::AutoScaling::AutoScalingGroup
   Properties:
      LaunchTemplate:
         LaunchTemplateId: !Ref 'LaunchTemplate'
         Version: !GetAtt 'LaunchTemplate.LatestVersionNumber'
      MaxSize: '1'
      MinSize: '0'
      DesiredCapacity: '1'
      LoadBalancerNames:
         - !Ref 'ElasticLoadBalancer'
      HealthCheckType: ELB
      VPCZoneIdentifier:
         - !Ref 'Subnet'
```
YAML example

```yaml
AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    LaunchTemplate:
      LaunchTemplateId: !Ref 'LaunchTemplate'
      Version: !GetAtt 'LaunchTemplate.LatestVersionNumber'
    MaxSize: '1'
    MinSize: '0'
    DesiredCapacity: '1'
    TargetGroupARNs:
      - !Ref 'ELBv2TargetGroup'
    HealthCheckType: ELB
    VPCZoneIdentifier:
      - !Ref 'Subnet'
```

CT.AUTOScaling.PR.4 rule specification

```
# Rule Specification

Rule Identifier: autoscaling_group_elb_healthcheck_required_check

Description:
This control checks whether your Auto Scaling groups that are associated with a load balancer are using Elastic Load Balancing health checks.

Reports on: AWS::AutoScaling::AutoScalingGroup

Evaluates:
AWS CloudFormation, AWS CloudFormation hook

Rule Parameters:
None

Scenarios:
Scenario: 1
  Given: The input document is an AWS CloudFormation or CloudFormation hook document
  And: The input document does not contain any Auto Scaling group
  Then: SKIP

Scenario: 2
  Given: The input document is an AWS CloudFormation or CloudFormation hook document
  And: The input document contains an Auto Scaling group resource
  And: 'LoadBalancerNames' or 'TargetGroupARNs' are not present on the Auto Scaling group resource or empty lists
```

584
Proactive controls

Scenario: 3
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an Auto Scaling group resource
And: 'LoadBalancerNames' or 'TargetGroupARNs' are present on the Auto Scaling group with at least one configuration
And: 'HealthCheckType' is not present
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an Auto Scaling group resource
And: 'LoadBalancerNames' or 'TargetGroupARNs' are present on the Auto Scaling group with at least one configuration
And: 'HealthCheckType' is present and set to a value other than 'ELB' (e.g. 'EC2')
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an Auto Scaling group resource
And: 'LoadBalancerNames' or 'TargetGroupARNs' are present on the Auto Scaling group with at least one configuration
And: 'HealthCheckType' is present and set to 'ELB'
Then: PASS

[CT.AUTOSCALING.PR.5] Require than an Amazon EC2 Auto Scaling group launch configuration does not have Amazon EC2 instances with public IP addresses

This control checks whether Amazon EC2 Auto Scaling groups have public IP addresses configured through Launch Configurations.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::AutoScaling::LaunchConfiguration
- **AWS CloudFormation guard rule:** CT.AUTOSCALING.PR.5 rule specification (p. 586)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.AUTOSCALING.PR.5 rule specification (p. 586)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.AUTOSCALING.PR.5 example templates (p. 588)

**Explanation**

Amazon EC2 instances in an Auto Scaling group launch configuration should not have an associated public IP address, except for in limited edge cases. Amazon EC2 instances should only be accessible from behind a load balancer instead of being directly exposed to the internet.

**Remediation for rule failure**

Set AssociatePublicIpAddress to false on Auto Scaling Launch Configurations.
The examples that follow show how to implement this remediation.

**Auto Scaling Launch Configuration - Example**

Auto Scaling Launch Configuration configured to disable public IP address association. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "AutoScalingLaunchConfiguration": {
      "Type": "AWS::AutoScaling::LaunchConfiguration",
      "Properties": {
         "ImageId": {
            "Ref": "LatestAmiId"
         },
         "InstanceType": "t3.micro",
         "AssociatePublicIpAddress": false
      }
   }
}
```

**YAML example**

```yaml
AutoScalingLaunchConfiguration:
   Type: AWS::AutoScaling::LaunchConfiguration
   Properties:
      ImageId: !Ref 'LatestAmiId'
      InstanceType: t3.micro
      AssociatePublicIpAddress: false
```

**CT.AUTOSCALING.PR.5 rule specification**

```plaintext
# ###################################
##       Rule Specification        ##
####################################
#
# Rule Identifier:
#   autoscaling_launch_config_public_ip_disabled_check
#
# Description:
#   Checks if Auto Scaling Launch Configurations have been configured to disable public IP address association.
#
# Reports on:
#   AWS::Auto Scaling::LaunchConfiguration
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation Hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#   Given: The input document is an AWS CloudFormation or CloudFormation hook document
```
# And: The input document does not contain any Auto Scaling Launch Configuration Resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Auto Scaling Launch Configuration Resource
# And: 'AssociatePublicIpAddress' is not present on the Auto Scaling Launch Configuration Resource
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Auto Scaling Launch Configuration Resource
# And: 'AssociatePublicIpAddress' is present on the Auto Scaling Launch Configuration Resource
# and is set to bool(true)
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Auto Scaling Launch Configuration Resource
# And: 'AssociatePublicIpAddress' is present on the Auto Scaling Launch Configuration Resource
# and is set to bool(false)
# Then: PASS

# Constants
# let AUTOSCALING_LAUNCH_CONFIGURATION_TYPE = 'AWS::AutoScaling::LaunchConfiguration'
let INPUT_DOCUMENT = this

# Assignments
# let autoscaling_launch_configurations = Resources.*[ Type == %AUTOSCALING_LAUNCH_CONFIGURATION_TYPE ]

# Primary Rules
# rule autoscaling_launch_config_public_ip_disabled_check when
# is_cfn_template(%INPUT_DOCUMENT)
%autoscaling_launch_configurations not empty {
  check(%autoscaling_launch_configurations.Properties)
  <<
    [CT.AUTOSCALING.PR.5]: Require than an Amazon EC2 Auto Scaling group launch configuration does not have EC2 instances with public IP addresses
    [FIX]: Set 'AssociatePublicIpAddress' to false on Auto Scaling Launch Configurations.
  >>
}

rule autoscaling_launch_config_public_ip_disabled_check when is_cfn_hook(%INPUT_DOCUMENT, %AUTOSCALING_LAUNCH_CONFIGURATION_TYPE) {
  check(%INPUT_DOCUMENT.%AUTOSCALING_LAUNCH_CONFIGURATION_TYPE.resourceProperties)
  <<
    [CT.AUTOSCALING.PR.5]: Require than an Amazon EC2 Auto Scaling group launch configuration does not have EC2 instances with public IP addresses
    [FIX]: Set 'AssociatePublicIpAddress' to false on Auto Scaling Launch Configurations.
  >>
}
CT.AUTOSCALING.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Parameters:
LatestAmiId:
  Description: Region specific latest AMI ID from the Parameter Store
  Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
AutoScalingLaunchConfiguration:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
    ImageId:
      Ref: LatestAmiId
    InstanceType: t3.micro
    AssociatePublicIpAddress: false

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Parameters:
LatestAmiId:
  Description: Region specific latest AMI ID from the Parameter Store
  Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
AutoScalingLaunchConfiguration:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
Proactive controls

[CT.AUTOSCALING.PR.6] Require any Amazon EC2 Auto Scaling groups to use multiple instance types

This control checks whether an Amazon EC2 Auto Scaling group uses multiple instance types through a mixed instance policy and explicit instance type overrides.

- **Control objective:** Improve availability
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::AutoScaling::AutoScalingGroup
- **AWS CloudFormation guard rule:** CT.AUTOSCALING.PR.6 rule specification (p. 590)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.AUTOSCALING.PR.6 rule specification (p. 590)
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: CT.AUTOSCALING.PR.6 example templates (p. 595)

**Explanation**

You can enhance availability by deploying your application across multiple instance types running in multiple Availability Zones. AWS Control Tower recommends using multiple instance types so that the Auto Scaling group can launch another instance type if there is insufficient instance capacity in your chosen Availability Zones.

**Usage considerations**

- This control applies only to Amazon EC2 Auto Scaling groups that do not use attribute-based instance type selection within a mixed instances policy (configured by means of the InstanceRequirements property within mixed instances policy Overrides).

**Remediation for rule failure**

Within a MixedInstancePolicy configuration, provide a LaunchTemplate configuration with two entries in the Overrides property. Within each override, set the InstanceType property to a different Amazon EC2 instance type.

The examples that follow show how to implement this remediation.

**Amazon EC2 Auto Scaling Group - Example**

Amazon EC2 Auto Scaling group configured with multiple instance types. The example is shown in JSON and in YAML.

**JSON example**

```json
ImageId:
   Ref: LatestAmiId
InstanceType: t3.micro
AssociatePublicIpAddress: true
```

589
YAML example

AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - !Ref 'Subnet'
    MaxSize: '2'
    MinSize: '1'
    MixedInstancesPolicy:
      LaunchTemplate:
        LaunchTemplateSpecification:
          LaunchTemplateId: !Ref 'EC2LaunchTemplate'
          Version: !GetAtt 'EC2LaunchTemplate.LatestVersionNumber'
          Overrides:
            - InstanceType: t3.micro
            - InstanceType: m5.large
Rule Identifier:
autoscaling_mixed_instances_policy_multiple_instance_types_check

Description:
This control checks whether an Amazon EC2 Auto Scaling group uses multiple instance types through a mixed instance policy and explicit instance type overrides.

Reports on:
AWS::AutoScaling::AutoscalingGroup

Evaluates:
AWS CloudFormation, AWS CloudFormation hook

Rule Parameters:
None

Scenarios:
Scenario: 1
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document does not contain any Autoscaling Group resources
Then: SKIP

Scenario: 2
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains Autoscaling Group resources
And: 'MixedInstancesPolicy.LaunchTemplate.Overrides' has been provided as a list
And: There exists any 'Overrides' entry where 'InstanceRequirements' is present
And: There exists no 'Overrides' entry where 'InstanceType' is present
Then: SKIP

Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains Autoscaling Group resources
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has been provided
And: 'MixedInstancesPolicy.LaunchTemplate.Overrides' has not been provided
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains Autoscaling Group resources
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has been provided
And: 'MixedInstancesPolicy.LaunchTemplate.Overrides' has been provided as a list
And: 'InstanceType' is not present or is present as a empty string in
'MixedInstancesPolicy.LaunchTemplate.Overrides'
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains Autoscaling Group resources
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has been provided
And: 'MixedInstancesPolicy.LaunchTemplate.Overrides' has been provided as a list
And: There exists any 'Overrides' entry where 'InstanceRequirements' is present
And: There exists any 'Overrides' entry where 'InstanceType' is present
Then: FAIL

Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains Autoscaling Group resources
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has been provided
And: 'MixedInstancesPolicy.LaunchTemplate.Overrides' has been provided as a list
And: 'InstanceType' is present in 'MixedInstancesPolicy.LaunchTemplate.Overrides' as a non empty string
And: Length of 'MixedInstancesPolicy.LaunchTemplate.Overrides' is less than or equal to 1
Then: FAIL

Scenario: 7
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains Autoscaling Group resources
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has been provided
And: 'MixedInstancesPolicy.LaunchTemplate.Overrides' has been provided as a list
And: 'InstanceType' is present in 'MixedInstancesPolicy.LaunchTemplate.Overrides' as a non empty string
And: Length of 'MixedInstancesPolicy.LaunchTemplate.Overrides' is greater than 1
Then: PASS

Constants
let AUTOSCALING_GROUP_TYPE = "AWS::AutoScaling::AutoScalingGroup"
let INPUT_DOCUMENT = this

Assignments
let autoscaling_groups = Resources.*[ Type == %AUTOSCALING_GROUP_TYPE ]

Primary Rules
rule autoscaling_mixed_instances_policy_multiple_instance_types_check when
  is_cfn_template(%INPUT_DOCUMENT)
  %autoscaling_groups not empty {
    check(%autoscaling_groups.Properties)
    <<
    [CT.AUTO_SCALING.PR.6]: Require any Amazon EC2 Auto Scaling groups to use multiple instance types
    [FIX]: Within a 'MixedInstancePolicy' configuration, provide a 'LaunchTemplate' configuration with two entries in the 'Overrides' property. Within each override, set the 'InstanceType' property to a different Amazon EC2 instance type.
    >>
  }

rule autoscaling_mixed_instances_policy_multiple_instance_types_check when
  is_cfn_hook(%INPUT_DOCUMENT, %AUTOSCALING_GROUP_TYPE) {
  check(%INPUT_DOCUMENT.%AUTOSCALING_GROUP_TYPE.resourceProperties)
  <<
  [CT.AUTO_SCALING.PR.6]: Require any Amazon EC2 Auto Scaling groups to use multiple instance types
  [FIX]: Within a 'MixedInstancePolicy' configuration, provide a 'LaunchTemplate' configuration with two entries in the 'Overrides' property. Within each override, set the 'InstanceType' property to a different Amazon EC2 instance type.
  >>
}

Parameterized Rules
rule check(autoscaling_group) {
  %autoscaling_group [
    # Scenario 2
filter_asg_no_instance_requirement_overrides(this)

} {  
 # Scenario 4, 5, 6
 MixedInstancesPolicy exists
 MixedInstancesPolicy is_struct

 MixedInstancesPolicy {
   LaunchTemplate exists
   LaunchTemplate is_struct

   LaunchTemplate {
     LaunchTemplateSpecification exists
     LaunchTemplateSpecification is_struct

     Overrides exists
     Overrides is_list
     Overrides not empty

     Overrides[0] exists
     Overrides[1] exists

     Overrides[*] {
       InstanceType exists
       check_is_string_and_not_empty(InstanceType)
     }

     Overrides[0].InstanceType not in Overrides[1].InstanceType
   }

 } }

%autoscaling_group [  
 # Scenario 2
 filter_asg_conflicting_overrides(this)  
] {  
 MixedInstancesPolicy {
   LaunchTemplate {
     Overrides[*] {
       check_mutually_exclusive_property_combination(InstanceType,
       InstanceRequirements) or
       check_mutually_exclusive_property_combination(InstanceRequirements,
       InstanceType)
     }
   }
 }

}

rule filter_asg_no_instance_requirement_overrides(autoscaling_group) {
  %autoscaling_group {
    MixedInstancesPolicy not exists or
    filter_mixed_instances_policy_no_instance_requirement_overrides(this)
  }
}

rule filter_mixed_instances_policy_no_instance_requirement_overrides(autoscaling_group) {
  %autoscaling_group {
    MixedInstancesPolicy is_struct
    MixedInstancesPolicy {
      LaunchTemplate not exists or
      filter_launch_templates_no_instance_requirement_overrides(this)
    }
  }
}

rule filter_launch_templates_no_instance_requirement_overrides(launch_template) {

}
%launch_template {
  LaunchTemplate is_struct
  LaunchTemplate {
    Overrides not exists or
    filter_overrides_no_instance_requirement_overrides(this)
  }
}

rule filter_overrides_no_instance_requirement_overrides(overrides) {
  %overrides {
    Overrides is_list
    Overrides empty or
    Overrides[*] {
      InstanceRequirements not exists
    }
  }
}

rule filter_asg_conflicting_overrides(autoscaling_group) {
  %autoscaling_group {
    MixedInstancesPolicy not exists or
    filter_mixed_instances_policy_conflicting_overrides(this)
  }
}

rule filter_mixed_instances_policy_conflicting_overrides(autoscaling_group) {
  %autoscaling_group {
    MixedInstancesPolicy is_struct
    MixedInstancesPolicy {
      LaunchTemplate not exists or
      filter_launch_templates_conflicting_overrides(this)
    }
  }
}

rule filter_launch_templates_conflicting_overrides(launch_template) {
  %launch_template {
    LaunchTemplate is_struct
    LaunchTemplate {
      Overrides not exists or
      filter_overrides_conflicting_overrides(this)
    }
  }
}

rule filter_overrides_conflicting_overrides(overrides) {
  %overrides {
    Overrides is_list
    Overrides empty or
    some Overrides[*] {
      InstanceRequirements exists
      InstanceType exists
    }
  }
}

rule check_mutually_exclusive_property_combination(property1, property2) {
  %property1 not exists
  %property2 exists
}

#    # Utility Rules
#    rule is_cfn_template(doc) {

CT.AUTOSCALING.PR.6 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Parameters:
- LatestAmiId:
  - Description: Region specific latest AMI ID from the Parameter Store
  - Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  - Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
- EC2LaunchTemplate:
  - Type: AWS::EC2::LaunchTemplate
  - Properties:
    - LaunchTemplateName:
      - Fn::Sub: ${AWS::StackName}-example
    - LaunchTemplateData:
      - InstanceType: t3.micro
      - ImageId:
        - Ref: LatestAmiId

- VPC:
  - Type: AWS::EC2::VPC
  - Properties:
    - CidrBlock: 10.0.0.0/16
    - EnableDnsSupport: 'true'
    - EnableDnsHostnames: 'true'

- Subnet:
  - Type: AWS::EC2::Subnet
  - Properties:
    - VpcId:
      - Ref: VPC
    - CidrBlock: 10.0.0.0/24
    - AvailabilityZone:
      - Fn::Select:
        - 0
        - Fn::GetAZs: '

- AutoScalingGroup:
  - Type: AWS::AutoScaling::AutoScalingGroup
  - Properties:
    - VPCZoneIdentifier:
      - Ref: Subnet
    - MaxSize: '2'
    - MinSize: '1'
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateId:
        Ref: EC2LaunchTemplate
      Version:
        Fn::GetAtt:
          - EC2LaunchTemplate
          - LatestVersionNumber
      Overrides:
        - InstanceType: t3.micro
        - InstanceType: m5.large

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Parameters:
  LatestAmiId:
    Description: Region specific latest AMI ID from the Parameter Store
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
  EC2LaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName:
        Fn::Sub: ${AWS::StackName}-example
      LaunchTemplateData:
        InstanceType: t3.micro
        ImageId:
          Ref: LatestAmiId
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
  Subnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/24
      AvailabilityZone:
        Fn::Select:
          - 0
          - Fn::GetAZs: ''

AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - Ref: Subnet
    MaxSize: '2'
    MinSize: '1'
    MixedInstancesPolicy:
      LaunchTemplate:
        LaunchTemplateSpecification:
          LaunchTemplateId:
            Ref: EC2LaunchTemplate
          Version:
            Fn::GetAtt:
              - EC2LaunchTemplate
              - LatestVersionNumber
[CT.AUTOSCALING.PR.8] Require an Amazon EC2 Auto Scaling group to have EC2 launch templates configured

This control checks whether an Amazon EC2 Auto Scaling group is configured to use an EC2 launch template.

- **Control objective:** Manage vulnerabilities
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::AutoScaling::AutoScalingGroup
- **AWS CloudFormation guard rule:** CT.AUTOSCALING.PR.8 rule specification (p. 599)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.AUTOSCALING.PR.8 rule specification (p. 599)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.AUTOSCALING.PR.8 example templates (p. 603)

**Explanation**

An Auto Scaling group can be created from an EC2 launch template or from a launch configuration. If you use a launch template to create an Auto Scaling group, you have access to the latest features and improvements.

**Remediation for rule failure**

Provide a LaunchTemplate or MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification configuration with a valid Version and a LaunchTemplateId or LaunchTemplateName.

The examples that follow show how to implement this remediation.

**Amazon EC2 Auto Scaling Group - Example One**

Amazon EC2 Auto Scaling group configured with an EC2 launch template. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "AutoScalingGroup": {
      "Type": "AWS::AutoScaling::AutoScalingGroup",
      "Properties": {
         "VPCZoneIdentifier": [
            {
               "Ref": "Subnet"
            }
         ]
      }
   }
}
```
"MaxSize": "2",
"MinSize": "1",
"LaunchTemplate": {
  "LaunchTemplateName": "SampleLaunchTemplate",
  "Version": {
    "Fn::GetAtt": [
      "EC2LaunchTemplate",
      "LatestVersionNumber"
    ]
  }
}
}

YAML example

AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - !Ref 'Subnet'
    MaxSize: '2'
    MinSize: '1'
    LaunchTemplate:
      LaunchTemplateName: SampleLaunchTemplate
      Version: !GetAtt 'EC2LaunchTemplate.LatestVersionNumber'

The examples that follow show how to implement this remediation.

Amazon EC2 Auto Scaling Group - Example Two

Amazon EC2 Auto Scaling group configured with a mixed instances policy and EC2 launch template. The example is shown in JSON and in YAML.

JSON example

```json
{
  "AutoScalingGroup": {
    "Type": "AWS::AutoScaling::AutoScalingGroup",
    "Properties": {
      "VPCZoneIdentifier": [
        {
          "Ref": "Subnet"
        }
      ],
      "MaxSize": "2",
      "MinSize": "1",
      "MixedInstancesPolicy": {
        "LaunchTemplate": {
          "LaunchTemplateSpecification": {
            "LaunchTemplateId": {
              "Ref": "EC2LaunchTemplate"
            },
            "Version": {
              "Fn::GetAtt": [
                "EC2LaunchTemplate",
                "LatestVersionNumber"
              ]
            }
          }
        }
      }
    }
  }
}
```
YAML example

AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
Properties:
  VPCZoneIdentifier:
    - !Ref 'Subnet'
  MaxSize: '2'
  MinSize: '1'
  MixedInstancesPolicy:
    LaunchTemplate:
      LaunchTemplateSpecification:
        LaunchTemplateId: !Ref 'EC2LaunchTemplate'
        Version: !GetAtt 'EC2LaunchTemplate.LatestVersionNumber'

CT.AUTOSCALING.PR.8 rule specification

# ####################################################################
##       Rule Specification        
# ####################################################################
#
# Rule Identifier:
#   autoscaling_launch_template_check
#
# Description:
#   This control checks whether an Amazon EC2 Auto Scaling group is configured to use an
#   EC2 launch template.
# # Reports on:
#   AWS::AutoScaling::AutoscalingGroup
# # Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# # Rule Parameters:
#   None
# # Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#   document
#   And: The input document does not contains any Amazon EC2 Auto Scaling group
#   resources
#   Then: SKIP
# Scenario: 2
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#   document
#   And: The input document contains an Amazon EC2 Auto Scaling group resource
And: 'LaunchTemplate' has not been provided
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has not been provided
Then: FAIL

Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon EC2 Auto Scaling group resource
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has not been provided
And: 'LaunchTemplate' has been provided
And: 'LaunchTemplate' has an invalid configuration ('Version' has not been provided and one of 'LaunchTemplateId' or 'LaunchTemplateName' has not been provided or 'Version' has been provided as an empty string or invalid local reference and one of 'LaunchTemplateId' or 'LaunchTemplateName' has been provided as an empty string or an invalid local reference)
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon EC2 Auto Scaling group resource
And: 'LaunchTemplate' has not been provided
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has been provided
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has an invalid configuration ('Version' has not been provided and one of 'LaunchTemplateId' or 'LaunchTemplateName' has not been provided or 'Version' has been provided as an empty string or invalid local reference and one of 'LaunchTemplateId' or 'LaunchTemplateName' has been provided as an empty string or an invalid local reference)
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon EC2 Auto Scaling group resource
And: 'LaunchTemplate' has been provided
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has been provided.
Then: FAIL

Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon EC2 Auto Scaling group resource
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has not been provided
And: 'LaunchTemplate' has been provided
And: 'LaunchTemplate' has a valid configuration ('Version' has been provided and one of 'LaunchTemplateId' or 'LaunchTemplateName' has been provided as a non-empty string or valid local reference)
Then: PASS

Scenario: 7
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon EC2 Auto Scaling group resource
And: 'LaunchTemplate' has not been provided
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has been provided
And: 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' has a valid configuration ('Version' has been provided and one of 'LaunchTemplateId' or 'LaunchTemplateName' has been provided as a non-empty string or valid local reference)
# Constants

let AUTOSCALING_GROUP_TYPE = "AWS::AutoScaling::AutoScalingGroup"
let INPUT_DOCUMENT = this

# Assignments

let autoscaling_groups = Resources.*[ Type == %AUTOSCALING_GROUP_TYPE ]

# Primary Rules

rule autoscaling_launch_template_check when is_cfn_template(%INPUT_DOCUMENT) { %autoscaling_groups not empty {
    check(%autoscaling_groups.Properties) <<
    [CT.AUTOSCALING.PR.8]: Require an Amazon EC2 Auto Scaling group to have EC2 launch templates configured
    [FIX]: Provide a 'LaunchTemplate' or 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' configuration with a valid 'Version' and a 'LaunchTemplateId' or 'LaunchTemplateName'.
    >>
}

rule autoscaling_launch_template_check when is_cfn_hook(%INPUT_DOCUMENT, %AUTOSCALING_GROUP_TYPE) {
    check(%INPUT_DOCUMENT.%AUTOSCALING_GROUP_TYPE.resourceProperties) <<
    [CT.AUTOSCALING.PR.8]: Require an Amazon EC2 Auto Scaling group to have EC2 launch templates configured
    [FIX]: Provide a 'LaunchTemplate' or 'MixedInstancesPolicy.LaunchTemplate.LaunchTemplateSpecification' configuration with a valid 'Version' and a 'LaunchTemplateId' or 'LaunchTemplateName'.
    >>
}

# Parameterized Rules

rule check(autoscaling_groups) {
    %autoscaling_groups {
        # Scenario 3 and 6
        check_launch_template(this) or
        # Scenario 4 and 7
        check_mixed_instances_policy(this)
    }
}

rule check_launch_template(autoscaling_groups) {
    %autoscaling_groups {
        check_mutually_exclusive_property_combination(LaunchTemplate, MixedInstancesPolicy)
        LaunchTemplate is_struct
        LaunchTemplate {
            check_valid_launch_template_config(this)
        }
    }
}

rule check_mixed_instances_policy(autoscaling_groups) {
    %autoscaling_groups {
        check_mutually_exclusive_property_combination(MixedInstancesPolicy, LaunchTemplate)
AWS Control Tower User Guide
Proactive controls

}

}

MixedInstancesPolicy is_struct
MixedInstancesPolicy {
LaunchTemplate exists
LaunchTemplate is_struct
LaunchTemplate {
LaunchTemplateSpecification exists
LaunchTemplateSpecification is_struct
check_valid_launch_template_config(LaunchTemplateSpecification)
}
}

rule check_valid_launch_template_config(launch_template_specification) {
%launch_template_specification {
check_valid_launch_template_property(Version)
check_valid_prop_combination(LaunchTemplateId, LaunchTemplateName) or
check_valid_prop_combination(LaunchTemplateName, LaunchTemplateId)
}
}
rule check_valid_prop_combination(valid_property, invalid_property) {
check_mutually_exclusive_property_combination(%valid_property, %invalid_property)
check_valid_launch_template_property(%valid_property)
}
rule check_mutually_exclusive_property_combination(valid_property, invalid_property) {
%invalid_property not exists
%valid_property exists
}
rule check_valid_launch_template_property(property) {
%property {
check_is_string_and_not_empty(this) or
check_local_references(%INPUT_DOCUMENT, this, "AWS::EC2::LaunchTemplate")
}
}
rule check_is_string_and_not_empty(value) {
%value {
this is_string
this != /\A\s*\z/
}
}
#
# Utility Rules
#
rule is_cfn_template(doc) {
%doc {
AWSTemplateFormatVersion exists
Resources exists
}
}

or

rule is_cfn_hook(doc, RESOURCE_TYPE) {
%doc.%RESOURCE_TYPE.resourceProperties exists
}
rule check_local_references(doc, reference_properties, referenced_resource_type) {
%reference_properties {
'Fn::GetAtt' {
query_for_resource(%doc, this[0], %referenced_resource_type)
<<Local Stack reference was invalid>>
} or Ref {
query_for_resource(%doc, this, %referenced_resource_type)

602


CT.AUTOSCALING.PR.8 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Parameters:
LatestAmiId:
  Description: Region specific latest AMI ID from the Parameter Store
  Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
EC2LaunchTemplate:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName:
      Fn::Sub: ${AWS::StackName}-example
    LaunchTemplateData:
      InstanceType: t3.micro
      ImageId:
        Ref: LatestAmiId

VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'

Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: ''

AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - Ref: Subnet
    MaxSize: '2'
    MinSize: '1'
    LaunchTemplate:
      LaunchTemplateId:
        Ref: EC2LaunchTemplate
    Version:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Parameters:
- LatestAmiId:
  Description: Region specific latest AMI ID from the Parameter Store
  Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Resources:
- VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
- Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      - Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone: 
      - Fn::Select:
        - 0
        - Fn::GetAZs: ''
- AutoScalingLaunchConfiguration:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
    ImageId:
      - Ref: LatestAmiId
    InstanceType: t3.micro
    LaunchConfigurationName: "AutoScalingLaunchConfiguration"
- AutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    VPCZoneIdentifier:
      - Ref: Subnet
    MaxSize: '2'
    MinSize: '1'
    LaunchConfigurationName: "AutoScalingLaunchConfiguration"

Amazon ElastiCache controls

Topics
- [CT.ELASTICACHE.PR.1] Require an Amazon ElastiCache for Redis cluster to have automatic backups activated (p. 605)
- [CT.ELASTICACHE.PR.3] Require an Amazon ElastiCache for Redis replication group to have automatic failover activated (p. 610)
- [CT.ELASTICACHE.PR.4] Require an Amazon ElastiCache replication group to have encryption at rest activated (p. 615)
- [CT.ELASTICACHE.PR.5] Require an Amazon ElastiCache for Redis replication group to have encryption in transit activated (p. 619)
[CT.ELASTICACHE.PR.6] Require an Amazon ElastiCache cache cluster to use a custom subnet group (p. 624)

[CT.ELASTICACHE.PR.7] Require an Amazon ElastiCache replication group of earlier Redis versions to have Redis AUTH activated (p. 630)

[CT.ELASTICACHE.PR.1] Require an Amazon ElastiCache for Redis cluster to have automatic backups activated

This control checks whether an Amazon ElastiCache Redis cluster has automatic backups enabled.

- **Control objective:** Improve resiliency
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElastiCache::CacheCluster
- **AWS CloudFormation guard rule:** [CT.ELASTICACHE.PR.1 rule specification](p. 606)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICACHE.PR.1 rule specification](p. 606)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.ELASTICACHE.PR.1 example templates](p. 608)

**Explanation**

When automatic backups are enabled, Amazon ElastiCache creates a backup of the cluster on a daily basis. There is no impact on the cluster, and the change is immediate. Automatic backups can help guard against data loss. In the event of a failure, you can create a new cluster, and restore your data from the most recent backup.

**Usage considerations**

- This control applies only to Amazon ElastiCache cache clusters with an engine type of redis

**Remediation for rule failure**

Set the value of the SnapshotRetentionLimit parameter to an integer value greater than 0.

The examples that follow show how to implement this remediation.

**Amazon ElastiCache Cache Cluster - Example**

An Amazon ElastiCache cache cluster configured with automatic backups enabled. The example is shown in JSON and in YAML.

**JSON example**

```
{
    "CacheCluster": {
        "Type": "AWS::ElastiCache::CacheCluster",
        "Properties": {
            "Engine": "redis",
            "NumCacheNodes": 1,
```
YAML example

```yaml
CacheCluster:
  Type: AWS::ElastiCache::CacheCluster
  Properties:
    Engine: redis
    NumCacheNodes: 1
    CacheNodeType: cache.t3.micro
    VpcSecurityGroupIds:
      - !Ref 'SecurityGroup'
    CacheSubnetGroupName: !Ref 'SubnetGroup'
    SnapshotRetentionLimit: 5
```

CT.ELASTICACHE.PR.1 rule specification

```bash
# #############################################################################
#                    Rule Specification                                      #
# #############################################################################
#
# Rule Identifier:    elasticache_redis_cluster_auto_backup_check
# Description:        This control checks whether an Amazon ElastiCache Redis cluster has automatic backups enabled.
# Reports on:         AWS::ElastiCache::CacheCluster
# Evaluates:          AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:    None
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any ElastiCache cache cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
```
And: The input document contains an ElastiCache CacheCluster resource
And: 'Engine' has not been provided or has been provided and is not set to 'redis'
Then: SKIP
Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ElastiCache cache cluster resource
And: 'Engine' has been provided and set to 'redis'
And: 'SnapshotRetentionLimit' has not been provided
Then: FAIL
Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ElastiCache cache cluster resource
And: 'Engine' has been provided and set to 'redis'
And: 'SnapshotRetentionLimit' has been provided and set to a non-integer value or an integer value of 0
Then: FAIL
Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ElastiCache cache cluster resource
And: 'Engine' has been provided and set to 'redis'
And: 'SnapshotRetentionLimit' has been provided and set to an integer value greater than 0
Then: PASS

# Constants
let ELASTICACHE_CACHE_CLUSTER_TYPE = "AWS::ElastiCache::CacheCluster"
let REDIS_ENGINE_TYPE = "redis"
let INPUT_DOCUMENT = this

# Assignments
let elasticache_clusters = Resources.*[ Type == %ELASTICACHE_CACHE_CLUSTER_TYPE ]

# Primary Rules
rule elasticache_redis_cluster_auto_backup_check when is_cfn_template(%INPUT_DOCUMENT) {  
  check(%elasticache_clusters.Properties) {  
    %elasticache_clusters not empty {  
      [CT.ELASTICACHE.PR.1]: Require an Amazon ElastiCache for Redis cluster to have automatic backups activated  
      [FIX]: Set the value of the 'SnapshotRetentionLimit' parameter to an integer value greater than 0.  
    }  
  }  
}
rule elasticache_redis_cluster_auto_backup_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICACHE_CACHE_CLUSTER_TYPE) {  
  check(%INPUT_DOCUMENT.%ELASTICACHE_CACHE_CLUSTER_TYPE.resourceProperties) {  
    %ELASTICACHE_CACHE_CLUSTER_TYPE.resourceProperties not empty {  
      [CT.ELASTICACHE.PR.1]: Require an Amazon ElastiCache for Redis cluster to have automatic backups activated  
      [FIX]: Set the value of the 'SnapshotRetentionLimit' parameter to an integer value greater than 0.  
    }  
  }  
}

# Parameterized Rules
# Proactive controls

```python
# rule check(elasticache_cache_cluster) {
  %elasticache_cache_cluster [
    # Scenario 2
    Engine exists
    Engine == %REDIS_ENGINE_TYPE
  ] {
    # Scenarios 3, 4 and 5
    SnapshotRetentionLimit exists
    SnapshotRetentionLimit > 0
  }
}
#
# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

CT.ELASTICACHE.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
  Subnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/16
      AvailabilityZone:
        Fn::Select:
          - 0
          - Fn::GetAZs: ''
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: test
      VpcId:
        Ref: VPC
      SecurityGroupIngress:
        - FromPort: 443
          IpProtocol: tcp
          ToPort: 443
          CidrIp: 0.0.0.0/0
  SubnetGroup:
    Type: AWS::ElastiCache::SubnetGroup
```

608
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Properties:
- Description: Cache Subnet Group
  SubnetIds:
  - Ref: Subnet

CacheCluster:
Type: AWS::ElastiCache::CacheCluster
Properties:
- Engine: redis
  NumCacheNodes: 1
  CacheNodeType: cache.t3.micro
  VpcSecurityGroupIds:
    - Ref: SecurityGroup
  CacheSubnetGroupName:
    Ref: SubnetGroup
  SnapshotRetentionLimit: 5

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16

Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/16
  AvailabilityZone:
    Fn::Select:
      0
    - Fn::GetAZs: '

SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: test
    VpcId:
      Ref: VPC
    SecurityGroupIngress:
      - FromPort: 443
        IpProtocol: tcp
        ToPort: 443
        CidrIp: 0.0.0.0/0

SubnetGroup:
  Type: AWS::ElastiCache::SubnetGroup
  Properties:
    Description: Cache Subnet Group
    SubnetIds:
      - Ref: Subnet

CacheCluster:
  Type: AWS::ElastiCache::CacheCluster
  Properties:
    Engine: redis
    NumCacheNodes: 1
    CacheNodeType: cache.t3.micro
    VpcSecurityGroupIds:
      - Ref: SecurityGroup
    CacheSubnetGroupName:
      Ref: SubnetGroup
    SnapshotRetentionLimit: 0
[CT.ELASTICACHE.PR.3] Require an Amazon ElastiCache for Redis replication group to have automatic failover activated

This control checks whether an Amazon ElastiCache Redis replication group has automatic failover enabled.

- **Control objective**: Improve resiliency
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::ElastiCache::ReplicationGroup
- **AWS CloudFormation guard rule**: CT.ELASTICACHE.PR.3 rule specification (p. 611)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ELASTICACHE.PR.3 rule specification (p. 611)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.ELASTICACHE.PR.3 example templates (p. 613)

**Explanation**

When automatic failover is enabled for a replication group, the role of primary node will fail over to one of the read replicas, automatically. This failover and replica promotion ensure that you can resume writing to the new primary as soon as promotion is complete, thereby reducing overall downtime in case of failure.

**Remediation for rule failure**

Set the value of the AutomaticFailoverEnabled parameter to true.

The examples that follow show how to implement this remediation.

**Amazon ElastiCache Replication Group - Example**

An Amazon ElastiCache replication group configured with automatic failover enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ReplicationGroup": {
        "Type": "AWS::ElastiCache::ReplicationGroup",
        "Properties": {
            "ReplicationGroupDescription": "Sample replication group",
            "CacheNodeType": "cache.t3.micro",
            "SecurityGroupIds": [
                {
                    "Ref": "SecurityGroup"
                }
            ],
            "CacheSubnetGroupName": {
                "Ref": "SubnetGroup"
            }
        }
    }
}
```
YAML example

ReplicationGroup:
  Type: AWS::ElastiCache::ReplicationGroup
  Properties:
    ReplicationGroupDescription: Sample replication group
    CacheNodeType: cache.t3.micro
    SecurityGroupIds:
      - !Ref 'SecurityGroup'
    CacheSubnetGroupName: !Ref 'SubnetGroup'
    NumCacheClusters: 2
    Engine: redis
    AutomaticFailoverEnabled: true

CT.ELASTICACHE.PR.3 rule specification

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
## Rule Specification
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Rule Identifier:
# elasticache_repl_grp_backup_enabled_check
#
# Description:
# This control checks whether an Amazon ElastiCache Redis replication group has automatic
# failover enabled.
#
# Reports on:
# AWS::ElastiCache::ReplicationGroup
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
# document
#   And: The input document does not contain any ElastiCache replication group
#   resources
#   Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
# document
#   And: The input document contains an ElastiCache replication group resource
#   And: 'Engine' has not been provided or has been provided and is not set to 'redis'
#   Then: SKIP
# Scenario: 3

611
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElastiCache replication group resource
# And: 'Engine' has been provided and set to 'redis'
# And: 'AutomaticFailoverEnabled' has not been provided
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElastiCache replication group resource
# And: 'Engine' has been provided and set to 'redis'
# And: 'AutomaticFailoverEnabled' has been provided and is set to a value other than bool(true)
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElastiCache replication group resource
# And: 'Engine' has been provided and set to 'redis'
# And: 'AutomaticFailoverEnabled' has been provided and is set to a value of bool(true)
# Then: PASS

# Constants

let ELASTICACHE_REPLICATION_GROUP_TYPE = "AWS::ElastiCache::ReplicationGroup"
let REDIS_ENGINE_TYPE = "redis"
let INPUT_DOCUMENT = this

# Assignments

let elasticache_replication_groups = Resources.*[ Type == %ELASTICACHE_REPLICATION_GROUP_TYPE ]

# Primary Rules

rule elasticache_repl_grp_auto_failover_enabled_check when is_cfn_template(%INPUT_DOCUMENT) %elasticache_replication_groups not empty {
    check(%elasticache_replication_groups.Properties)
    // [CT.ELASTICACHE.PR.3]: Require an Amazon ElastiCache for Redis replication group to have automatic failover activated
    [FIX]: Set the value of the 'AutomaticFailoverEnabled' parameter to true.
    //
}

rule elasticache_repl_grp_auto_failover_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICACHE_REPLICATION_GROUP_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTICACHE_REPLICATION_GROUP_TYPE.resourceProperties)
    // [CT.ELASTICACHE.PR.3]: Require an Amazon ElastiCache for Redis replication group to have automatic failover activated
    [FIX]: Set the value of the 'AutomaticFailoverEnabled' parameter to true.
    //
}

# # Parameterized Rules
# # rule check(elasticache_replication_group) {
#    %elasticache_replication_group [ # Scenario 2

# Proactive controls

Engine exists
Engine == %REDIS_ENGINE_TYPE

} {
    # Scenarios 3, 4 and 5
    AutomaticFailoverEnabled exists
    AutomaticFailoverEnabled == true
}

# Utility Rules

## CT.ELASTICACHE.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```
Resources:
VPC:
    Type: AWS::EC2::VPC
    Properties:
        CidrBlock: 10.0.0.0/16
Subnet:
    Type: AWS::EC2::Subnet
    Properties:
        VpcId:
            Ref: VPC
        CidrBlock: 10.0.0.0/16
        AvailabilityZone:
            Fn::Select:
            - 0
            - Fn::GetAZs: ''
SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
        GroupDescription: Example security group
        VpcId:
            Ref: VPC
        SecurityGroupIngress:
            - FromPort: 443
              IpProtocol: tcp
              ToPort: 443
              CidrIp: 0.0.0.0/0
SubnetGroup:
    Type: AWS::ElastiCache::SubnetGroup
    Properties:
        Description: Example subnet group
        SubnetIds:
            - Ref: Subnet
```
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
  Subnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: Ref: VPC
      CidrBlock: 10.0.0.0/16
      AvailabilityZone:
        Fn::Select:
        - 0
        - Fn::GetAZs: ''
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Example security group
      VpcId:
        Ref: VPC
      SecurityGroupIngress:
        - FromPort: 443
          IpProtocol: tcp
          ToPort: 443
          CidrIp: 0.0.0.0/0
  SubnetGroup:
    Type: AWS::ElastiCache::SubnetGroup
    Properties:
      Description: Example subnet group
      SubnetIds:
        - Ref: Subnet
  ReplicationGroup:
    Type: AWS::ElastiCache::ReplicationGroup
    Properties:
      ReplicationGroupDescription:
        Fn::Sub: '${AWS::StackName}-example'
      CacheNodeType: cache.t3.micro
      SecurityGroupIds:
        - Ref: SecurityGroup
      CacheSubnetGroupName:
        Ref: SubnetGroup
      NumCacheClusters: 2
      Engine: redis
      AutomaticFailoverEnabled: true
[CT.ELASTICACHE.PR.4] Require an Amazon ElastiCache replication group to have encryption at rest activated

This control checks whether an Amazon ElastiCache replication group has the encryption-at-rest setting enabled.

- **Control objective**: Encrypt data at rest
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::ElastiCache::ReplicationGroup
- **AWS CloudFormation guard rule**: CT.ELASTICACHE.PR.4 rule specification (p. 616)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ELASTICACHE.PR.4 rule specification (p. 616)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.ELASTICACHE.PR.4 example templates (p. 618)

**Explanation**

Encryption of data at rest is a recommended best practice that adds a layer of access management around your data. In case of any compromise to your Amazon ElastiCache replica nodes, this encryption-at-rest setting ensures that your data is protected from unintended access.

**Usage considerations**

- This control requires the use of encryption at rest, which is supported only for replication groups with Redis engine versions of 3.2.6 or above.

**Remediation for rule failure**

Set the value of the AtRestEncryptionEnabled parameter to true.

The examples that follow show how to implement this remediation.

**Amazon ElastiCache Replication Group - Example**

Amazon ElastiCache replication group configured with encryption at rest enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ElastiCacheReplicationGroup": {
        "Type": "AWS::ElastiCache::ReplicationGroup",
        "Properties": {
            "CacheSubnetGroupName": {
                "Ref": "SubnetGroup"
            },
            "CacheNodeType": "cache.t3.medium",
            "CacheParameterGroupName": "default.cache_1.2015-01-20",
            "Engine": "redis",
            "EngineVersion": "redis-5.0.5",
            "ReplicationGroupDescription": "My replication group"
        }
    }
}
```
"NumCacheClusters": 2,
"Engine": "redis",
"ReplicationGroupDescription": "Sample replication group",
"AtRestEncryptionEnabled": true
}
}

YAML example

ElastiCacheReplicationGroup:
Type: AWS::ElastiCache::ReplicationGroup
Properties:
  CacheSubnetGroupName: !Ref 'SubnetGroup'
  CacheNodeType: cache.t3.medium
  NumCacheClusters: 2
  Engine: redis
  ReplicationGroupDescription: Sample replication group
  AtRestEncryptionEnabled: true

CT.ELASTICACHE.PR.4 rule specification

# ###################################
##       Rule Specification        ##
####################################
#
# Rule Identifier:
#   elasticache_repl_grp_encrypted_at_rest_check
#
# Description:
#   This control checks whether an Amazon ElastiCache replication group has the encryption-at-rest setting enabled.
#
# Reports on:
#   AWS::ElastiCache::ReplicationGroup
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#  Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any ElastiCache ReplicationGroup resources
#     Then: SKIP
#  Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElastiCache ReplicationGroup resource
#     And: 'Engine' has not been provided or has been provided and is not set to 'redis'
#     Then: SKIP
#  Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElastiCache ReplicationGroup resource
#
And: 'Engine' has been provided and set to 'redis'
And: 'AtRestEncryptionEnabled' has not been provided
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ElastiCache ReplicationGroup resource
And: 'Engine' has been provided and set to 'redis'
And: 'AtRestEncryptionEnabled' has been provided and is set to a value other than bool(true)
Then: FAIL
Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ElastiCache ReplicationGroup resource
And: 'Engine' has been provided and set to 'redis'
And: 'AtRestEncryptionEnabled' has been provided and is set to a value of bool(true)
Then: PASS

Constants
let ELASTICACHE_REPLICATION_GROUP_TYPE = "AWS::ElastiCache::ReplicationGroup"
let INPUT_DOCUMENT = this

Assignments
let elasticache_replication_groups = Resources.*[ Type == %ELASTICACHE_REPLICATION_GROUP_TYPE ]

Primary Rules
rule elasticache_repl_grp_encrypted_at_rest_check when is_cfn_template(%INPUT_DOCUMENT)
%elasticache_replication_groups not empty {
  check(%elasticache_replication_groups.Properties):
<<
[CT.ELASTICACHE.PR.4]: Require an Amazon ElastiCache replication group to have encryption at rest activated
[FIX]: Set the value of the 'AtRestEncryptionEnabled' parameter to true.
>>
}

rule elasticache_repl_grp_encrypted_at_rest_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICACHE_REPLICATION_GROUP_TYPE) {
  check(%INPUT_DOCUMENT.%ELASTICACHE_REPLICATION_GROUP_TYPE.resourceProperties):
<<
[CT.ELASTICACHE.PR.4]: Require an Amazon ElastiCache replication group to have encryption at rest activated
[FIX]: Set the value of the 'AtRestEncryptionEnabled' parameter to true.
>>
}

Parameterized Rules
rule check(elasticache_replication_group) {
  %elasticache_replication_group [ Scenario 2
    filter_elasticache_replication_group(this)
  ] {
    # Scenario 3
    AtRestEncryptionEnabled exists
  }
# Scenarios 4 and 5
AtRestEncryptionEnabled == true  
}

rule filter_elasticache_replication_group(elasticache_replication_group) {
  %elasticache_replication_group {
    Engine exists
    Engine == "redis"
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

## CT.ELASTICACHE.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

Resources:

**VPC:**
  - Type: AWS::EC2::VPC
  - Properties:
    - CidrBlock: 10.0.0.0/16
    - EnableDnsSupport: 'true'
    - EnableDnsHostnames: 'true'

**Subnet:**
  - Type: AWS::EC2::Subnet
  - Properties:
    - VpcId: Ref: VPC
    - CidrBlock: 10.0.1.0/24
    - AvailabilityZone: Fn::Select:
      - 0
      - Fn::GetAZs: ''

**SubnetGroup:**
  - Type: AWS::ElastiCache::SubnetGroup
  - Properties:
    - Description: Example subnet group
    - SubnetIds:
      - Ref: Subnet

**ElastiCacheReplicationGroup:**
  - Type: AWS::ElastiCache::ReplicationGroup
  - Properties:
    - CacheSubnetGroupName:
      - Ref: SubnetGroup
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone: 
      Fn::Select:
        - 0
        - Fn::GetAZs: ''
SubnetGroup:
  Type: AWS::ElastiCache::SubnetGroup
  Properties:
    Description: Example subnet group
    SubnetIds:
      - Ref: Subnet
ElastiCacheReplicationGroup:
  Type: AWS::ElastiCache::ReplicationGroup
  Properties:
    CacheSubnetGroupName:
      Ref: SubnetGroup
    CacheNodeType: cache.t3.medium
    NumCacheClusters: 2
    Engine: redis
    ReplicationGroupDescription: Example replication group
    AtRestEncryptionEnabled: false

[CT.ELASTICACHE.PR.5] Require an Amazon ElastiCache for Redis replication group to have encryption in transit activated

This control checks whether an Amazon ElastiCache replication group has encryption-in-transit enabled.

- **Control objective:** Encrypt data in transit
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElastiCache::ReplicationGroup
- **AWS CloudFormation guard rule:** [CT.ELASTICACHE.PR.5 rule specification](p. 621)

Details and examples
• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICACHE.PR.5 rule specification (p. 621)]

• For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.ELASTICACHE.PR.5 example templates (p. 623)]

Explanation

TLS can be used to help prevent potential attackers from eavesdropping on or manipulating network traffic using person-in-the-middle or similar attacks. Amazon ElastiCache in-transit encryption is an optional feature that you can use to help protect your data when it is moving from one location to another.

Usage considerations

• Encryption-in-transit is supported on Amazon ElastiCache replication groups running Redis versions 3.2.6, 4.0.10 and later.

• Because of the processing required to encrypt and decrypt the data at the endpoints, implementing in-transit encryption can reduce performance. We recommend that you benchmark in-transit encryption, compared to no encryption, on your own data, to determine the impact of encryption-in-transit on performance for your implementation.

Remediation for rule failure

Set the value of the TransitEncryptionEnabled parameter to true.

The examples that follow show how to implement this remediation.

Amazon ElastiCache Replication Group - Example

Amazon ElastiCache replication group configured with encryption-in-transit enabled. The example is shown in JSON and in YAML.

JSON example

```json
{
  "ReplicationGroup": {
    "Type": "AWS::ElastiCache::ReplicationGroup",
    "Properties": {
      "ReplicationGroupDescription": "Sample replication group",
      "CacheNodeType": "cache.t3.micro",
      "SecurityGroupIds": [
        {
          "Ref": "SecurityGroup"
        }
      ],
      "CacheSubnetGroupName": {
        "Ref": "SubnetGroup"
      },
      "NumCacheClusters": 2,
      "Engine": "redis",
      "TransitEncryptionEnabled": true
    }
  }
}
```

YAML example

```yaml
---
ReplicationGroup:
  Type: AWS::ElastiCache::ReplicationGroup
  Properties:
    ReplicationGroupDescription: Sample replication group
    CacheNodeType: cache.t3.micro
    SecurityGroupIds:
      - !Ref SecurityGroup
    CacheSubnetGroupName: !Ref SubnetGroup
    NumCacheClusters: 2
    Engine: redis
    TransitEncryptionEnabled: true
---
```
ReplicationGroup:
  Type: AWS::ElastiCache::ReplicationGroup
  Properties:
    - ReplicationGroupDescription: Sample replication group
    - CacheNodeType: cache.t3.micro
    - SecurityGroupIds: !Ref 'SecurityGroup'
    - CacheSubnetGroupName: !Ref 'SubnetGroup'
    - NumCacheClusters: 2
    - Engine: redis
    - TransitEncryptionEnabled: true

CT.ELASTICACHE.PR.5 rule specification

# ###################################################################
##       Rule Specification        ##
###################################################################
# 
# Rule Identifier:
#   elasticache_repl_grp_encrypted_in_transit_check
# 
# Description:
#   This control checks whether an Amazon ElastiCache replication group has encryption-in-
#   transit enabled.
# 
# Reports on:
#   AWS::ElastiCache::ReplicationGroup
# 
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# 
# Rule Parameters:
#   None
# 
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any ElastiCache ReplicationGroup resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an ElastiCache ReplicationGroup resource
#     And: 'Engine' has not been provided or has been provided and is not set to 'redis'
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an ElastiCache ReplicationGroup resource
#     And: 'Engine' has been provided and set to 'redis'
#     And: 'TransitEncryptionEnabled' has not been provided
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an ElastiCache ReplicationGroup resource
#     And: 'Engine' has been provided and set to 'redis'
#     And: 'TransitEncryptionEnabled' has been provided and set to a value other than
#     bool(true)
# Scenario 5

*Given:* The input document is an AWS CloudFormation or AWS CloudFormation hook document
*And:* The input document contains an ElastiCache ReplicationGroup resource
*And:* 'Engine' has been provided and set to 'redis'
*And:* 'TransitEncryptionEnabled' has been provided and set to bool(true)
*Then:* PASS

---

# Constants

let ELASTICACHE_REPLICATION_GROUP_TYPE = "AWS::ElastiCache::ReplicationGroup"
let REDIS_ENGINE_TYPE = "redis"
let INPUT_DOCUMENT = this

# Assignments

let elasticache_replication_groups = Resources.*[ Type == %ELASTICACHE_REPLICATION_GROUP_TYPE ]

# Primary Rules

rule elasticache_repl_grp_encrypted_in_transit_check when is_cfn_template(%INPUT_DOCUMENT) 
%elasticache_replication_groups 
not empty {
  check(%elasticache_replication_groups.Properties) 
  <<
      [CT.ELASTICACHE.PR.5]: Require an Amazon ElastiCache for Redis replication group to have encryption in transit activated
      [FIX]: Set the value of the 'TransitEncryptionEnabled' parameter to true.
  >>
}

rule elasticache_repl_grp_encrypted_in_transit_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICACHE_REPLICATION_GROUP_TYPE) {
  check(%INPUT_DOCUMENT.%ELASTICACHE_REPLICATION_GROUP_TYPE.resourceProperties) 
  <<
      [CT.ELASTICACHE.PR.5]: Require an Amazon ElastiCache for Redis replication group to have encryption in transit activated
      [FIX]: Set the value of the 'TransitEncryptionEnabled' parameter to true.
  >>
}

# Parameterized Rules

rule check(elasticache_replication_group) {
  %elasticache_replication_group [
    # Scenario 2
    Engine exists
    Engine == %REDIS_ENGINE_TYPE
  ] {
    # Scenarios 3, 4 and 5
    TransitEncryptionEnabled exists
    TransitEncryptionEnabled == true
  }
}

# Utility Rules

rule is_cfn_template(doc) {
  %doc {

CT.ELASTICACHE.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/16
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: ''
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Example security group
    VpcId:
      Ref: VPC
    SecurityGroupIngress:
      - FromPort: 443
        IpProtocol: tcp
        ToPort: 443
        CidrIp: 0.0.0.0/0
SubnetGroup:
  Type: AWS::ElastiCache::SubnetGroup
  Properties:
    Description: Example subnet group
    SubnetIds:
      - Ref: Subnet
ReplicationGroup:
  Type: AWS::ElastiCache::ReplicationGroup
  Properties:
    ReplicationGroupDescription:
      Fn::Sub: ${AWS::StackName}-example
    CacheNodeType: cache.t3.micro
    SecurityGroupIds:
      - Ref: SecurityGroup
    CacheSubnetGroupName:
      Ref: SubnetGroup
    NumCacheClusters: 2
    Engine: redis
    TransitEncryptionEnabled: true
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:

VPC:
- Type: AWS::EC2::VPC
- Properties:
  - CidrBlock: 10.0.0.0/16

Subnet:
- Type: AWS::EC2::Subnet
- Properties:
  - VpcId: Ref: VPC
  - CidrBlock: 10.0.0.0/16
  - AvailabilityZone:
    - Fn::Select:
      - 0
      - Fn::GetAZs: 

SecurityGroup:
- Type: AWS::EC2::SecurityGroup
- Properties:
  - GroupDescription: Example security group
  - VpcId: Ref: VPC
  - SecurityGroupIngress:
    - FromPort: 443
    - IpProtocol: tcp
    - ToPort: 443
    - CidrIp: 0.0.0.0/0

SubnetGroup:
- Type: AWS::ElastiCache::SubnetGroup
- Properties:
  - Description: Example subnet group
  - SubnetIds:
    - Ref: Subnet

ReplicationGroup:
- Type: AWS::ElastiCache::ReplicationGroup
- Properties:
  - ReplicationGroupDescription:
    - Fn::Sub: ${AWS::StackName}-example
  - CacheNodeType: cache.t3.micro
  - SecurityGroupIds:
    - Ref: SecurityGroup
  - CacheSubnetGroupName: Ref: SubnetGroup
  - NumCacheClusters: 2
  - Engine: redis
  - TransitEncryptionEnabled: false

[CT.ELASTICACHE.PR.6] Require an Amazon ElastiCache cache cluster to use a custom subnet group

This control checks whether an Amazon ElastiCache cache cluster is configured with a custom subnet group.

- **Control objective**: Limit network access
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
• **Resource types:** AWS::ElastiCache::CacheCluster
• **AWS CloudFormation guard rule:** [CT.ELASTICACHE.PR.6 rule specification](p. 626)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICACHE.PR.6 rule specification](p. 626)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.ELASTICACHE.PR.6 example templates](p. 628)

**Explanation**

When you launch an ElastiCache cluster, AWS creates a default subnet group if none exists already. The default group utilizes subnets from the default VPC. Using custom subnet groups allows you to be more restrictive about network access to ElastiCache clusters.

**Usage considerations**

- This rule evaluates whether an Amazon ElastiCache cache cluster has been configured with a custom subnet group.
- Custom subnet groups may contain subnets that reside in the default Amazon VPC.

**Remediation for rule failure**

Set the `CacheSubnetGroupName` parameter to the name of a custom Amazon ElastiCache cache subnet group.

The examples that follow show how to implement this remediation.

**Amazon ElastiCache Cache Cluster - Example**

An Amazon ElastiCache cache cluster configured with a custom subnet group. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ElasticacheCluster": {
    "Type": "AWS::ElastiCache::CacheCluster",
    "Properties": {
      "Engine": "memcached",
      "CacheNodeType": "cache.t3.micro",
      "NumCacheNodes": "1",
      "CacheSubnetGroupName": {
        "Ref": "SubnetGroup"
      },
      "VpcSecurityGroupIds": [
        {
          "Fn::GetAtt": [
            "SecurityGroup",
            "GroupId"
          ]
        }
      ]
    }
  }
}
```
YAML example

ElasticacheCluster:
  Type: AWS::ElastiCache::CacheCluster
  Properties:
    Engine: memcached
    CacheNodeType: cache.t3.micro
    NumCacheNodes: '1'
    CacheSubnetGroupName: !Ref 'SubnetGroup'
    VpcSecurityGroupIds:
      - !GetAtt 'SecurityGroup.GroupId'

CT.ELASTICACHE.PR.6 rule specification

# #######################################################################
#         Rule Specification       #
# #######################################################################
#
# Rule Identifier:
#   elasticache_subnet_group_check
#
# Description:
#   This control checks whether an Amazon ElastiCache cache cluster is configured with a
#   custom subnet group.
#
# Reports on:
#   AWS::ElastiCache::CacheCluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#        document
#     And: The input document does not contain any ElastiCache cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#        document
#     And: The input document contains an ElastiCache cluster resource
#     And: 'CacheSubnetGroupName' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#        document
#     And: The input document contains an ElastiCache cluster resource
#     And: 'CacheSubnetGroupName' has been provided as an empty string or as a non-valid
#        local reference
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#        document
#     And: The input document contains an ElastiCache cluster resource

# Scenario 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ElastiCache cluster resource
And: 'CacheSubnetGroupName' has been provided as a non-empty string or a valid local reference
Then: PASS

# Constants

let ELASTICACHE_CACHE_CLUSTER_TYPE = "AWS::ElastiCache::CacheCluster"
let INPUT_DOCUMENT = this

# Assignments

let elasticache_cache_clusters = Resources.*[ Type == %ELASTICACHE_CACHE_CLUSTER_TYPE ]

# Primary Rules

rule elasticache_subnet_group_check when is_cfn_template(%INPUT_DOCUMENT)
%elasticache_cache_clusters not empty {
  check(%elasticache_cache_clusters.Properties)
  <<
  [CT.ELASTICACHE.PR.6]: Require an Amazon ElastiCache cache cluster to use a custom subnet group
  [FIX]: Set the 'CacheSubnetGroupName' parameter to the name of a custom Amazon ElastiCache cache subnet group.
  >>
}

rule elasticache_subnet_group_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICACHE_CACHE_CLUSTER_TYPE) {
  check(%INPUT_DOCUMENT.%ELASTICACHE_CACHE_CLUSTER_TYPE.resourceProperties)
  <<
  [CT.ELASTICACHE.PR.6]: Require an Amazon ElastiCache cache cluster to use a custom subnet group
  [FIX]: Set the 'CacheSubnetGroupName' parameter to the name of a custom Amazon ElastiCache cache subnet group.
  >>
}

# Parameterized Rules

rule check(elasticache_cache_cluster) {
  %elasticache_cache_cluster {
    # Scenario 2
    CacheSubnetGroupName exists

    # Scenarios 3, 4 and 5
    check_subnet_group_is_not_default(this) or
    check_local_references(%INPUT_DOCUMENT, CacheSubnetGroupName, "AWS::ElastiCache::SubnetGroup")
  }
}

rule check_subnet_group_is_not_default(elasticache_cache_cluster) {
  %elasticache_cache_cluster {
    check_is_string_and_not_empty(CacheSubnetGroupName)
    CacheSubnetGroupName != "default"
  }
}
CT.ELASTICACHE.PR.6 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Example security group
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 11211
        ToPort: 11211
        CidrIp: 10.0.0.0/24
CacheCluster:
  Type: AWS::ElastiCache::CacheCluster
  Properties:
    Engine: memcached
    CacheNodeType: cache.t3.micro
    NumCacheNodes: '1'
    CacheSubnetGroupName: Ref: SubnetGroup
    VpcSecurityGroupIds:
      - Fn::GetAtt: [SecurityGroup, GroupId]

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Example security group
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 11211
        ToPort: 11211
        CidrIp: 10.0.0.0/24
CacheCluster:
  Type: AWS::ElastiCache::CacheCluster
  Properties:
    Engine: memcached
    CacheNodeType: cache.t3.micro
    NumCacheNodes: '1'
    CacheSubnetGroupName: Ref: SubnetGroup
    VpcSecurityGroupIds:
      - Fn::GetAtt: [SecurityGroup, GroupId]
[CT.ELASTICACHE.PR.7] Require an Amazon ElastiCache replication group of earlier Redis versions to have Redis AUTH activated

This control checks whether an Amazon ElastiCache replication group with an engine version earlier than 6.0 has Redis AUTH enabled.

- **Control objective**: Enforce least privilege
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::ElastiCache::ReplicationGroup
- **AWS CloudFormation guard rule**: CT.ELASTICACHE.PR.7 rule specification (p. 631)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ELASTICACHE.PR.7 rule specification (p. 631)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.ELASTICACHE.PR.7 example templates (p. 634)

Explanation

Redis authentication tokens, or passwords, enable Redis to require a password before allowing clients to run commands, thereby improving data security.

**Usage considerations**

- This control applies only to Amazon ElastiCache replication groups of Redis engine versions earlier than six (6).
- This control requires encryption-in-transit to be enabled on replication groups by means of the TransitEncryptionEnabled property.

Remediation for rule failure

Set the value of the AuthToken parameter to a string between 16 characters and 128 characters in length, which contains only printable ASCII characters and does not contain non-alphanumeric characters outside of the set (!, &,
The examples that follow show how to implement this remediation.

**Amazon ElastiCache Replication Group - Example**

An Amazon ElastiCache replication group configured with Redis AUTH authentication enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ReplicationGroup": {
    "Type": "AWS::ElastiCache::ReplicationGroup",
    "Properties": {
      "ReplicationGroupDescription": "Sample replication group",
      "CacheNodeType": "cache.t3.micro",
      "SecurityGroupIds": [
        { "Ref": "SecurityGroup" }
      ],
      "CacheSubnetGroupName": { "Ref": "SubnetGroup" },
      "NumCacheClusters": 2,
      "Engine": "redis",
      "EngineVersion": "5.0.6",
      "TransitEncryptionEnabled": true,
      "AuthToken": {
        "Fn::Sub": "{{resolve:secretsmanager:${ReplicationGroupSecret}::password}}"
      }
    }
  }
}
```

**YAML example**

```
ReplicationGroup:
  Type: AWS::ElastiCache::ReplicationGroup
  Properties:
    ReplicationGroupDescription: Sample replication group
    CacheNodeType: cache.t3.micro
    SecurityGroupIds: 
      - !Ref 'SecurityGroup'
    CacheSubnetGroupName: !Ref 'SubnetGroup'
    NumCacheClusters: 2
    Engine: redis
    EngineVersion: 5.0.6
    TransitEncryptionEnabled: true
    AuthToken: !Sub '{{resolve:secretsmanager:${ReplicationGroupSecret}::password}}'
```

**CT.ELASTICACHE.PR.7 rule specification**

```
# ####################################################################
#       Rule Specification       #
# ####################################################################
```

631
# Rule Identifier:
#   elasticache_repl_grp_redis_auth_enabled_check
#
# Description:
#   This control checks whether an Amazon ElastiCache replication group with an engine
#   version earlier than 6.0 has Redis AUTH enabled.
#
# Reports on:
#   AWS::ElastiCache::ReplicationGroup
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#            And: The input document does not contain any ElastiCache replication group
#            resources
#            Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#            And: The input document contains an ElastiCache replication group resource
#            And: 'Engine' has not been provided or has been provided and is not set to 'redis'
#            Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#            And: The input document contains an ElastiCache replication group resource
#            And: 'Engine' has been provided and is set to 'redis'
#            And: 'EngineVersion' has not been provided or has been provided and set to a
#            version greater than or equal to 6
#            Then: SKIP
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#            And: The input document contains an ElastiCache replication group resource
#            And: 'Engine' has been provided and is set to 'redis'
#            And: 'EngineVersion' has been provided and set to a version less than 6
#            And: 'AuthToken' has not been provided or has been provided and set to an empty
#            string
#            Then: FAIL
#   Scenario: 5
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#            And: The input document contains an ElastiCache replication group resource
#            And: 'Engine' has been provided and is set to 'redis'
#            And: 'EngineVersion' has been provided and set to a version less than 6
#            And: 'AuthToken' has been provided and set to a non-empty string
#            Then: PASS
#
# Constants
#
# let ELASTICACHE_REPLICATION_GROUP_TYPE = "AWS::ElastiCache::ReplicationGroup"
# let REDIS_ENGINE_TYPE = "redis"
# let SUPPORTED_REDIS_ENGINE_VERSIONS_FOR_REDIS_AUTH = [
#   /^2\./,
#   /^3\./,
#   /^4\./,
#   /^5\./
# ]
let INPUT_DOCUMENT = this

# # Assignments
let elasticache_replication_groups = Resources.*[ Type == %ELASTICACHE_REPLICATION_GROUP_TYPE ]

# # Primary Rules
# rule elasticache_repl grp redis auth enabled check when is_cfn_template(%INPUT_DOCUMENT)
empty { %elasticache_replication_groups not empty { check(%elasticache_replication_groups.Properties)
<<
[CT.ELASTICACHE.PR.7]: Require an Amazon ElastiCache replication group of earlier Redis versions to have Redis AUTH activated
[FIX]: Set the value of the 'AuthToken' parameter to a string between 16 characters and 128 characters in length, which contains only printable ASCII characters and does not contain non-alphanumeric characters outside of the set (!, &, >>
}

rule elasticache_replgrp_redistro_auth_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICACHE_REPLICATION_GROUP_TYPE) {
check(%INPUT_DOCUMENT.%ELASTICACHE_REPLICATION_GROUP_TYPE.resourceProperties)
<<
[CT.ELASTICACHE.PR.7]: Require an Amazon ElastiCache replication group of earlier Redis versions to have Redis AUTH activated
[FIX]: Set the value of the 'AuthToken' parameter to a string between 16 characters and 128 characters in length, which contains only printable ASCII characters and does not contain non-alphanumeric characters outside of the set (!, &, >>
}

# # Parameterized Rules
# rule check(elasticache_replication_group) {
%elasticache_replication_group [
# Scenario 2
Engine exists
Engine == %REDIS_ENGINE_TYPE

# Scenario 3
EngineVersion exists
EngineVersion in %SUPPORTED_REDIS_ENGINE_VERSIONS_FOR_REDIS_AUTH
] {
# Scenarios 4 and 5
AuthToken exists
check_is_string_and_not_empty(AuthToken)
}

# # Utility Rules
# rule is_cfn_template(doc) {
%doc {
AWSTemplateFormatVersion exists or Resources exists
}
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {

```
%doc.%RESOURCE_TYPE.resourceProperties exists
}

rule check_is_string_and_not_empty(value) {
  %value {
    this is_string
    this != /\A\s*\z/
  }
}

CT.ELASTICACHE.PR.7 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/16
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: ''
    SecurityGroup:
      Type: AWS::EC2::SecurityGroup
      Properties:
        GroupDescription: Example security group
        VpcId:
          Ref: VPC
        SecurityGroupIngress:
          - FromPort: 443
            IpProtocol: tcp
            ToPort: 443
            CidrIp: 0.0.0.0/0
      SubnetGroup:
        Type: AWS::ElastiCache::SubnetGroup
        Properties:
          Description: Cache subnet group
          SubnetIds:
            - Ref: Subnet
  ReplicationGroupSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: Replication group secret
      GenerateSecretString:
        SecretStringTemplate: "{}"
        GenerateStringKey: password
        PasswordLength: 64
        ExcludePunctuation: true
      ReplicationGroup:
        Type: AWS::ElastiCache::ReplicationGroup
        Properties:
          ReplicationGroupDescription:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/16
    AvailabilityZone:
      Fn::Select:
        - 0
      - Fn::GetAZs: ''
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Example security group
    VpcId:
      Ref: VPC
    SecurityGroupIngress:
      - FromPort: 443
        IpProtocol: tcp
        ToPort: 443
        CidrIp: 0.0.0.0/0
SubnetGroup:
  Type: AWS::ElastiCache::SubnetGroup
  Properties:
    Description: Cache subnet group
    SubnetIds:
      - Ref: Subnet
ReplicationGroup:
  Type: AWS::ElastiCache::ReplicationGroup
  Properties:
    ReplicationGroupDescription:
      Fn::Sub: '${AWS::StackName}-example'
    CacheNodeType: cache.t3.micro
    SecurityGroupIds:
      - Ref: SecurityGroup
    CacheSubnetGroupName:
      Ref: SubnetGroup
    NumCacheClusters: 2
    Engine: redis
    EngineVersion: 3.2.6
    TransitEncryptionEnabled: true
    AuthToken:
      Fn::Sub: '{[resolve:secretsmanager:${ReplicationGroupSecret}::password]}''
Amazon Elastic Container Registry controls

Topics

- [CT.ECR.PR.1] Require Amazon ECR repositories to have a lifecycle policy configured (p. 636)
- [CT.ECR.PR.2] Require Amazon ECR private repositories to have image scanning enabled (p. 640)
- [CT.ECR.PR.3] Require Amazon ECR private repositories to have tag immutability enabled (p. 643)

[CT.ECR.PR.1] Require Amazon ECR repositories to have a lifecycle policy configured

This control checks whether a private Amazon Elastic Container Registry (Amazon ECR) repository has at least one lifecycle policy configured.

- **Control objective:** Manage vulnerabilities, Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ECR::Repository
- **AWS CloudFormation guard rule:** CT.ECR.PR.1 rule specification (p. 637)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ECR.PR.1 rule specification (p. 637)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.ECR.PR.1 example templates (p. 639)

Explanation

Amazon ECR lifecycle policies specify the lifecycle management of images in a repository. By configuring lifecycle policies, you can automate the cleanup of unused images and the expiration of images, based on age or count. Automating these tasks can help you to avoid unintentionally using outdated images in your repository.

Remediation for rule failure

Provide a LifecyclePolicy configuration and set LifecyclePolicyText to an Amazon ECR repository lifecycle policy.

The examples that follow show how to implement this remediation.

Amazon ECR Repository - Example

Amazon ECR repository configured with a lifecycle policy. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ECRRepository": {
        "Type": "AWS::ECR::Repository",
```
"Properties": {
    "LifecyclePolicy": {
        "LifecyclePolicyText": "{\n        "rules": [{
            "rulePriority": 1,
            "description": "Expire images older than 14 days",
            "selection": {
                "tagStatus": "untagged",
                "countType": "sinceImagePushed",
                "countUnit": "days",
                "countNumber": 14
            },
            "action": {
                "type": "expire"
            }
        }],
        "type": "expire"
    }
}

YAML example

ECRRepository:
  Type: AWS::ECR::Repository
  Properties:
    LifecyclePolicy:
      LifecyclePolicyText: |
        
        
        
        
        "rules": [
          {
            "rulePriority": 1,
            "description": "Expire images older than 14 days",
            "selection": {
                "tagStatus": "untagged",
                "countType": "sinceImagePushed",
                "countUnit": "days",
                "countNumber": 14
            },
            "action": {
                "type": "expire"
            }
          }
        ]

CT.ECR.PR.1 rule specification

# ##################################################################
# Rule Specification  #
# ##################################################################
#
# Rule Identifier:
#    ecr_private_lifecycle_policy_configured_check
#
# Description:
#    This control checks whether a private Amazon Elastic Container Registry (ECR)
#    repository has at least one lifecycle policy configured.
#
# Reports on:
#    AWS::ECR::Repository
#
# Evaluates:
#    AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#    None
# Scenarios:
# Scenario: 1
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#    And: The input document does not contain any ECR repository resources
#    Then: SKIP
# Scenario: 2
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#    And: The input document contains an ECR repository resource
#    And: 'LifecyclePolicy' is not present
#    Then: FAIL
# Scenario: 3
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#    And: The input document contains an ECR repository resource
#    And: 'LifecyclePolicy' is present
#    And: 'LifecyclePolicyText' has not been provided in the 'LifecyclePolicy' configuration or has been provided as
#           an empty string
#    Then: FAIL
# Scenario: 4
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#    And: The input document contains an ECR repository resource
#    And: 'LifecyclePolicy' is present
#    And: 'LifecyclePolicyText' has been provided in the 'LifecyclePolicy' configuration with a non-empty string
#    Then: PASS

# Constants

let ECR_REPOSITORY_TYPE = "AWS::ECR::Repository"
let INPUT_DOCUMENT = this

# Assignments

let ecr_repositories = Resources.*[ Type == %ECR_REPOSITORY_TYPE ]

# Primary Rules

rule ecr_private_lifecycle_policy_configured_check when is_cfn_template(%INPUT_DOCUMENT)
    %ecr_repositories not empty {
        check(%ecr_repositories.Properties)
        <<
        [CT.ECR.PR.1]: Require Amazon ECR repositories to have a lifecycle policy configured
        [FIX]: Provide a 'LifecyclePolicy' configuration and set 'LifecyclePolicyText' to an Amazon ECR repository lifecycle policy.
        >>
    }

rule ecr_private_lifecycle_policy_configured_check when is_cfn_hook(%INPUT_DOCUMENT, %ECR_REPOSITORY_TYPE) {
    check(%INPUT_DOCUMENT.%ECR_REPOSITORY_TYPE.resourceProperties)
    <<
    [CT.ECR.PR.1]: Require Amazon ECR repositories to have a lifecycle policy configured
    [FIX]: Provide a 'LifecyclePolicy' configuration and set 'LifecyclePolicyText' to an Amazon ECR repository lifecycle policy.
    >>
}
rule check(ecr_repository) {
  %ecr_repository {
    # Scenario 3
    LifecyclePolicy exists
    LifecyclePolicy is struct
    LifecyclePolicy {
      # Scenario 4
      LifecyclePolicyText exists
      check_is_string_and_not_empty(LifecyclePolicyText)
    }
  }
}

# Utility Rules
#
# rule check_is_string_and_not_empty(value) {
#  %value {
#    this is_string
#    this != /\A\s*\z/
#  }
#}

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.ECR.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  ECRRepository:
    Type: AWS::ECR::Repository
    Properties:
      LifecyclePolicy:
        LifecyclePolicyText: |
          {
            "rules": [
              {
                "rulePriority": 1,
                "description": "Expire images older than 14 days",
                "selection": {
                  "tagStatus": "untagged",
                  "countType": "sinceImagePushed",
                  "countUnit": "days",
                  "countNumber": 14
                },
                "action": {
                  "type": "expire"
                }
              }
            ]
          }
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ECRRepository:
  Type: AWS::ECR::Repository
  Properties: {}

[CT.ECR.PR.2] Require Amazon ECR private repositories to have image scanning enabled

This control checks whether a private Amazon Elastic Container Registry (Amazon ECR) repository has image scanning enabled.

- Control objective: Manage vulnerabilities
- Implementation: AWS CloudFormation Guard Rule
- Control behavior: Proactive
- Resource types: AWS::ECR::Repository
- AWS CloudFormation guard rule: CT.ECR.PR.2 rule specification (p. 641)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ECR.PR.2 rule specification (p. 641)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.ECR.PR.2 example templates (p. 643)

Explanation

Amazon ECR image scanning helps with identifying software vulnerabilities in your container images. Amazon ECR uses the Common Vulnerabilities and Exposures (CVEs) database from the open-source Clair project, and it provides a list of scan findings. Enabling image scanning on Amazon ECR repositories adds a layer of verification regarding the integrity and safety of the images being stored.

Remediation for rule failure

Set ScanOnPush in ImageScanningConfiguration to true.

The examples that follow show how to implement this remediation.

Amazon ECR Repository - Example

Amazon ECR repository with image scanning enabled. The example is shown in JSON and in YAML.

JSON example

{
"ECRRepository": {
  "Type": "AWS::ECR::Repository",
  "Properties": {
    "ImageScanningConfiguration": {
      "ScanOnPush": true
    }
  }
}

YAML example

ECRRepository:
  Type: AWS::ECR::Repository
  Properties:
    ImageScanningConfiguration: 
      ScanOnPush: true

CT.ECR.PR.2 rule specification

# ###################################################################
#       Rule Specification        
# ###################################################################
#
# Rule Identifier:
#   ecr_private_image_scanning_enabled_check
#
# Description:
#   This control checks whether a private Amazon Elastic Container Registry (Amazon ECR) repository has image scanning enabled.
#
# Reports on:
#   AWS::ECR::Repository
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any Amazon ECR repository resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Amazon ECR repository resource
#     And: 'ImageScanningConfiguration.ScanOnPush' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Amazon ECR repository resource
#     And: 'ImageScanningConfiguration.ScanOnPush' has been provided and set to a value other than bool(true)
# Scenario 4

Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon ECR repository resource
And: 'ImageScanningConfiguration.ScanOnPush' has been provided and set to bool(true)
Then: PASS

# Constants

let ECR_REPOSITORY_TYPE = "AWS::ECR::Repository"
let INPUT_DOCUMENT = this

# Assignments

let ecr_repositories = Resources.*[ Type == %ECR_REPOSITORY_TYPE ]

# Primary Rules

rule ecr_private_image_scanning_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%ecr_repositories not empty {
  check(%ecr_repositories.Properties)
  <<
  [CT.ECR.PR.2]: Require Amazon ECR private repositories to have image scanning enabled
  [FIX]: Set 'ScanOnPush' in 'ImageScanningConfiguration' to 'true'.
  >>
}

rule ecr_private_image_scanning_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ECR_REPOSITORY_TYPE) {
  check(%INPUT_DOCUMENT.%ECR_REPOSITORY_TYPE.resourceProperties)
  <<
  [CT.ECR.PR.2]: Require Amazon ECR private repositories to have image scanning enabled
  [FIX]: Set 'ScanOnPush' in 'ImageScanningConfiguration' to 'true'.
  >>
}

# Parameterized Rules

rule check(ecr_repository) {
  %ecr_repository {
    # Scenario 2
    ImageScanningConfiguration exists
    ImageScanningConfiguration is_struct
    ImageScanningConfiguration {
      # Scenario 3 and 4
      ScanOnPush exists
      ScanOnPush == true
    }
  }
}

# Utility Rules

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
  }
}
CT.ECR.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  ECRRepository:
    Type: AWS::ECR::Repository
    Properties:
      ImageScanningConfiguration:
        ScanOnPush: true
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```yaml
Resources:
  ECRRepository:
    Type: AWS::ECR::Repository
    Properties:
      ImageScanningConfiguration:
        ScanOnPush: false
```

[CT.ECR.PR.3] Require Amazon ECR private repositories to have tag immutability enabled

This control checks whether a private Amazon Elastic Container Registry (Amazon ECR) repository has tag immutability enabled.

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ECR::Repository
- **AWS CloudFormation guard rule:** CT.ECR.PR.3 rule specification (p. 644)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ECR.PR.3 rule specification (p. 644)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.ECR.PR.3 example templates (p. 646)
**Explanation**

Amazon ECR tag immutability enables customers to rely on the descriptive tags of an image as a reliable mechanism that tracks and uniquely identifies images. An immutable tag is static, which means that each tag refers to a unique image. This tagging improves reliability and scalability, because the use of a static tag always results in the same image being deployed. When configured, tag immutability prevents the tags from being overridden, which reduces the attack surface.

**Remediation for rule failure**

Set `ImageTagMutability` to `IMMUTABLE`.

The examples that follow show how to implement this remediation.

**Amazon ECR Repository - Example**

Amazon ECR repository configured with immutable tags. The example is shown in JSON and in YAML.

**JSON example**

```
{
    "ECRRepository": {
        "Type": "AWS::ECR::Repository",
        "Properties": {
            "ImageTagMutability": "IMMUTABLE"
        }
    }
}
```

**YAML example**

```
ECRRepository:
  Type: AWS::ECR::Repository
  Properties:
    ImageTagMutability: IMMUTABLE
```

**CT.ECR.PR.3 rule specification**

```
# ####################################################################
##       Rule Specification       ##
####################################################################
#
# Rule Identifier:
#   ecr_private_tag_immutability_enabled_check
#
# Description:
#   This control checks whether a private Amazon Elastic Container Registry (Amazon ECR) repository has tag immutability enabled.
#
# Reports on:
#   AWS::ECR::Repository
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
```
# Proactive controls

## Rule Parameters:

None

## Scenarios:

### Scenario: 1

- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document does not contain any ECR repository resources
- Then: SKIP

### Scenario: 2

- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document contains an ECR repository resource
- And: 'ImageTagMutability' has not been provided
- Then: FAIL

### Scenario: 3

- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document contains an ECR repository resource
- And: 'ImageTagMutability' has been provided with a value of 'MUTABLE'
- Then: FAIL

### Scenario: 4

- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document contains an ECR repository resource
- And: 'ImageTagMutability' has been provided with a value of 'IMMUTABLE'
- Then: PASS

## Constants

- let ECR_REPOSITORY_TYPE = "AWS::ECR::Repository"
- let INPUT_DOCUMENT = this

## Assignments

- let ecr_repositories = Resources.*[ Type == %ECR_REPOSITORY_TYPE ]

## Primary Rules

### Rule: ecr_private_tag_immutability_enabled_check when is_cfn_template(%INPUT_DOCUMENT)

- check(%ecr_repositories.Properties)
  - ecr_repositories not empty
  - [CT.ECR.PR.3]: Require Amazon ECR private repositories to have tag immutability enabled
  - [FIX]: Set 'ImageTagMutability' to 'IMMUTABLE'.

### Rule: ecr_private_tag_immutability_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ECR_REPOSITORY_TYPE)

- check(%INPUT_DOCUMENT.%ECR_REPOSITORY_TYPE.resourceProperties)
  - [CT.ECR.PR.3]: Require Amazon ECR private repositories to have tag immutability enabled
  - [FIX]: Set 'ImageTagMutability' to 'IMMUTABLE'.

## Parameterized Rules
CT.ECR.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
ECRRepository:
  Type: AWS::ECR::Repository
  Properties:
    ImageTagMutability: IMMUTABLE

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ECRRepository:
  Type: AWS::ECR::Repository
  Properties:
    ImageTagMutability: MUTABLE

Amazon Elastic Container Service controls

Topics
- [CT.ECS.PR.1] Require Amazon ECS Fargate Services to run on the latest Fargate platform version (p. 647)
- [CT.ECS.PR.2] Require any Amazon ECS cluster to have container insights activated (p. 654)
- [CT.ECS.PR.3] Require any Amazon ECS task definition to specify a user that is not the root (p. 658)
- [CT.ECS.PR.4] Require Amazon ECS tasks to use 'awsvpc' networking mode (p. 662)
• [CT.ECS.PR.5] Require an active Amazon ECS task definition to have a logging configuration (p. 666)
• [CT.ECS.PR.6] Require Amazon ECS containers to allow read-only access to the root filesystem (p. 671)
• [CT.ECS.PR.7] Require an Amazon ECS task definition to have a specific memory usage limit (p. 676)
• [CT.ECS.PR.8] Require Amazon ECS task definitions to have secure networking modes and user definitions (p. 680)
• [CT.ECS.PR.9] Require Amazon ECS services not to assign public IP addresses automatically (p. 686)
• [CT.ECS.PR.10] Require that Amazon ECS task definitions do not share the host's process namespace (p. 692)
• [CT.ECS.PR.11] Require an Amazon ECS container to run as non-privileged (p. 697)
• [CT.ECS.PR.12] Require that Amazon ECS task definitions do not pass secrets as container environment variables (p. 701)

[CT.ECS.PR.1] Require Amazon ECS Fargate Services to run on the latest Fargate platform version

This control checks whether Amazon Elastic Container Service (Amazon ECS) Fargate services are configured to deploy using the LATEST platform version rather than a specified version number.

• Control objective: Manage vulnerabilities
• Implementation: AWS CloudFormation Guard Rule
• Control behavior: Proactive
• Resource types: AWS::ECS::Service
• AWS CloudFormation guard rule: CT.ECS.PR.1 rule specification (p. 649)

Details and examples

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ECS.PR.1 rule specification (p. 649)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.ECS.PR.1 example templates (p. 652)

Explanation

AWS Fargate platform versions refer to a specific runtime environment for Fargate task infrastructure, which is a combination of kernel and container runtime versions. New platform versions are released as the runtime environment evolves. For example, a new version may be released for kernel or operating system updates, new features, bug fixes, or security updates. Security updates and patches are deployed automatically for your Fargate tasks. If a security issue is found that affects a platform version, AWS patches the platform version.

Usage considerations

• This control only applies to Amazon ECS services with a LaunchType of FARGATE

Remediation for rule failure

When LaunchType is set to FARGATE, set the PlatformVersion property to LATEST or omit the PlatformVersion property (default: LATEST).

The examples that follow show how to implement this remediation.
Amazon ECS Service - Example One

Amazon ECS service configured to deploy using the LATEST platform version by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ECSService": {
    "Type": "AWS::ECS::Service",
    "Properties": {
      "Cluster": {
        "Ref": "ECSCluster"
      },
      "DesiredCount": 1,
      "TaskDefinition": {
        "Ref": "ECSTaskDefinition"
      },
      "NetworkConfiguration": {
        "AwsvpcConfiguration": {
          "AssignPublicIp": "DISABLED",
          "Subnets": [
            "Ref": "SubnetOne"
          ],
          "Ref": "SubnetTwo"
        }
      },
      "LaunchType": "FARGATE"
    }
  }
}
```

**YAML example**

```yaml
ECSService:
  Type: AWS::ECS::Service
  Properties:
    Cluster: !Ref 'ECSCluster'
    DesiredCount: 1
    TaskDefinition: !Ref 'ECSTaskDefinition'
    NetworkConfiguration:
      AwsvpcConfiguration:
        AssignPublicIp: DISABLED
        Subnets:
        - !Ref 'SubnetOne'
        - !Ref 'SubnetTwo'
    LaunchType: FARGATE
```

The examples that follow show how to implement this remediation.

Amazon ECS Service - Example Two

Amazon ECS service configured to deploy using the LATEST platform version by means of the PlatformVersion property. The example is shown in JSON and in YAML.
JSON example

```json
{
    "ECSService": {
        "Type": "AWS::ECS::Service",
        "Properties": {
            "Cluster": {
                "Ref": "ECSCluster"
            },
            "DesiredCount": 1,
            "TaskDefinition": {
                "Ref": "ECSTaskDefinition"
            },
            "NetworkConfiguration": {
                "AwsvpcConfiguration": {
                    "AssignPublicIp": "DISABLED",
                    "Subnets": [
                        {
                            "Ref": "SubnetOne"
                        },
                        {
                            "Ref": "SubnetTwo"
                        }
                    ]
                },
                "LaunchType": "FARGATE",
                "PlatformVersion": "LATEST"
            }
        }
    }
}
```

YAML example

```yaml
ECSService:
  Type: AWS::ECS::Service
  Properties:
    Cluster: !Ref 'ECSCluster'
    DesiredCount: 1
    TaskDefinition: !Ref 'ECSTaskDefinition'
    NetworkConfiguration:
      AwsvpcConfiguration:
        AssignPublicIp: DISABLED
        Subnets:
        - !Ref 'SubnetOne'
        - !Ref 'SubnetTwo'
    LaunchType: FARGATE
    PlatformVersion: LATEST
```

CT.ECS.PR.1 rule specification

```
# ###################################################################
##       Rule Specification        
# ###################################################################
#
# Rule Identifier:
#   ecs_fargate_latest_platform_version_check
```
# Description:
# This control checks whether Amazon Elastic Container Service (Amazon ECS) Fargate
# services are configured to deploy using the 'LATEST' platform version rather than a
# specified version number.
#
# Reports on:
# AWS::ECS::Service
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
#   And: The input document does not contain an Amazon ECS service resource
#       Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
#   And: The input document contains an Amazon ECS service resource
#       And: 'LaunchType' is not present
#       Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
#   And: The input document contains an Amazon ECS service resource
#       And: 'LaunchType' is present and not set to 'FARGATE'
#       Then: SKIP
# Scenario: 4
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
#   And: The input document contains an Amazon ECS service resource
#       And: 'LaunchType' is present and set to 'FARGATE'
#       And: 'PlatformVersion' is present and not set to 'LATEST'
#       Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
#   And: The input document contains an Amazon ECS service resource
#       And: 'LaunchType' is present and set to 'FARGATE'
#       And: 'PlatformVersion' is not present
#       Then: PASS
# Scenario: 6
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
#   And: The input document contains an Amazon ECS service resource
#       And: 'LaunchType' is present and set to 'FARGATE'
#       And: 'PlatformVersion' is set to 'LATEST'
#       Then: PASS
#
# Constants
#
let ECS_SERVICE_TYPE = "AWS::ECS::Service"
let INPUT_DOCUMENT = this
#
# Assignments
#
let ecs_services = Resources.[ Type == %ECS_SERVICE_TYPE ]
# Primary Rules

rule ecs_fargate_latest_platform_version_check when is_cfn_template(%INPUT_DOCUMENT)
%ecs_services not empty {
  check(%ecs_services.Properties)
  <<
  [CT.ECS.PR.1]: Require Amazon ECS Fargate Services to run on the latest Fargate platform version
  [FIX]: When 'LaunchType' is set to 'FARGATE', set the 'PlatformVersion' property to 'LATEST' or omit the 'PlatformVersion' property (default: 'LATEST').
  >>
}

rule ecs_fargate_latest_platform_version_check when is_cfn_hook(%INPUT_DOCUMENT, %ECS_SERVICE_TYPE) {
  check(%INPUT_DOCUMENT.%ECS_SERVICE_TYPE.resourceProperties)
  <<
  [CT.ECS.PR.1]: Require Amazon ECS Fargate Services to run on the latest Fargate platform version
  [FIX]: When 'LaunchType' is set to 'FARGATE', set the 'PlatformVersion' property to 'LATEST' or omit the 'PlatformVersion' property (default: 'LATEST').
  >>
}

# Parameterized Rules

rule check(ecs_service) {
  %ecs_service [ filter_launch_type_is_fargate(this) ]{
    # Scenario 5
    PlatformVersion not exists or
    # Scenario 4 and 6
    check_fargate_version_latest(PlatformVersion)
  }
}

rule filter_launch_type_is_fargate(ecs_service) {
  %ecs_service {
    # Scenario 2
    LaunchType exists
    LaunchType is_string
    # Scenario 3
    LaunchType == "FARGATE"
  }
}

rule check_fargate_version_latest(property) {
  %property {
    this is_string
    this == "LATEST"
  }
}

# Utility Rules

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
%doc.%RESOURCE_TYPE.resourceProperties exists
}

**CT.ECS.PR.1 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```yaml
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
  SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/24
      AvailabilityZone:
        Fn::Select:
          - 0
          - Fn::GetAZs: ''
  SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.1.0/24
      AvailabilityZone:
        Fn::Select:
          - 1
          - Fn::GetAZs: ''
  ECSCluster:
    Type: AWS::ECS::Cluster
    Properties:
      CapacityProviders:
        - FARGATE
  ECSTaskDefinition:
    Type: AWS::ECS::TaskDefinition
    Properties:
      ContainerDefinitions:
        - Essential: true
          Image: nginx:latest
          Name: SampleContainer
          Memory: '512'
          RequiresCompatibilities:
            - FARGATE
          NetworkMode: awsvpc
          Cpu: 256
  ECSService:
    Type: AWS::ECS::Service
    Properties:
      Cluster:
        Ref: ECSCluster
      DesiredCount: 0
      TaskDefinition:
```

652
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'

SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: '

SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
        - 1
        - Fn::GetAZs: '

ECSCluster:
  Type: AWS::ECS::Cluster
  Properties:
    CapacityProviders:
      - FARGATE

ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    ContainerDefinitions:
      - Essential: true
        Image: nginx:latest
        Name: SampleContainer
        Memory: 512
    RequiresCompatibilities:
      - FARGATE
    NetworkMode: awsvpc
    Cpu: 256

ECSService:
  Type: AWS::ECS::Service
  Properties:
    Cluster:
      Ref: ECSCluster
    DesiredCount: 0
    TaskDefinition:
[CT.ECS.PR.2] Require any Amazon ECS cluster to have container insights activated

This control checks whether your Amazon Elastic Container Service (Amazon ECS) clusters have container insights enabled.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** `AWS::ECS::Cluster`
- **AWS CloudFormation guard rule:** [CT.ECS.PR.2 rule specification](p. 655)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ECS.PR.2 rule specification](p. 655)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ECS.PR.2 example templates](p. 657)

**Explanation**

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon ECS clusters. Use CloudWatch container insights to collect, aggregate, and summarize metrics and logs from your containerized applications and microservices. CloudWatch automatically collects metrics for many resources, such as CPU, memory, disk, and network. The container insights capability also provides diagnostic information, such as container restart failures, which helps you isolate issues and resolve them quickly. You can set CloudWatch alarms on the metrics that container insights collects.

**Remediation for rule failure**

Enable container insights on Amazon ECS clusters with an entry in `ClusterSettings` that has `Name` set to `containerInsights` and `Value` set to `enabled`.

The examples that follow show how to implement this remediation.

**Amazon ECS Cluster - Example**

Amazon ECS cluster configured with container insights enabled. The example is shown in JSON and in YAML.

**JSON example**
YAML example

ECSCluster:
  Type: AWS::ECS::Cluster
  Properties:
    ClusterSettings:
      - Name: containerInsights
        Value: enabled

CT.ECS.PR.2 rule specification

# ###################################################################
##       Rule Specification       ##
# ###################################################################
# Rule Identifier:
#   ecs_container_insights_enabled_check
# Description:
#   This control checks whether your Amazon Elastic Container Service (Amazon ECS) clusters
#   have container insights enabled.
# Reports on:
#   AWS::ECS::Cluster
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
#   None
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation
#           hook document
#           And: The input document does not contain an Amazon ECS cluster resource
#           Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation
#           hook document
#           And: The input document contains an Amazon ECS cluster resource
#           And: 'ClusterSettings' property is not present or is an empty list
#           Then: FAIL
# Scenario: 3
# Scenario: 4
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
# And: The input document contains an Amazon ECS cluster resource
# And: 'ClusterSettings' property is present
# And: An entry with 'Name' set to 'containerInsights' is not present in
# 'ClusterSettings'
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
# And: The input document contains an Amazon ECS cluster resource
# And: 'ClusterSettings' property is present
# And: An entry with 'Name' set to 'containerInsights' is present in
# 'ClusterSettings' with a 'Value' not set
to 'enabled'
# Then: FAIL
# Then: PASS

# Constants
let ECS_CLUSTER_TYPE = "AWS::ECS::Cluster"
let INPUT_DOCUMENT = this

# Assignments
#
let ecs_clusters = Resources.*[ Type == %ECS_CLUSTER_TYPE ]

# Primary Rules
#
rule ecs_container_insights_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%ecs_clusters not empty { check(%ecs_clusters.Properties)
# [CT.ECS.PR.2]: Require any Amazon ECS cluster to have container insights activated
[FIX]: Enable container insights on Amazon ECS clusters with an entry in
'ClusterSettings' that has 'Name' set to 'containerInsights' and 'Value' set to 'enabled'.
} } rule ecs_container_insights_enabled_check when is_cfn_hook(%INPUT_DOCUMENT,
%ECS_CLUSTER_TYPE) { check(%INPUT_DOCUMENT.%ECS_CLUSTER_TYPE.resourceProperties)
# [CT.ECS.PR.2]: Require any Amazon ECS cluster to have container insights activated
[FIX]: Enable container insights on Amazon ECS clusters with an entry in
'ClusterSettings' that has 'Name' set to 'containerInsights' and 'Value' set to 'enabled'.
} } # Parameterized Rules
#
rule check(ecs_cluster) {
  %ecs_cluster {
    # Scenario 2
}
ClusterSettings exists
ClusterSettings is_list
ClusterSettings not empty

# Scenario 3, 4 and 5
some ClusterSettings[*] {
    Name exists
    Value exists

    Name is_string
    Value is_string

    Name == "containerInsights"
    Value == "enabled"
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.ECS.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
ECSCluster:
    Type: AWS::ECS::Cluster
    Properties:
        ClusterSettings:
            - Name: containerInsights
              Value: enabled

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ECSCluster:
    Type: AWS::ECS::Cluster
    Properties:
        ClusterSettings:
            - Name: containerInsights
              Value: disabled
[CT.ECS.PR.3] Require any Amazon ECS task definition to specify a user that is not the root

This control checks whether Amazon Elastic Container Service (ECS) task definitions run as a non-root user user within Amazon ECS containers.

- **Control objective:** Enforce least privilege, Manage vulnerabilities
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ECS::TaskDefinition
- **AWS CloudFormation guard rule:** [CT.ECS.PR.3 rule specification](p. 659)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ECS.PR.3 rule specification](p. 659)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ECS.PR.3 example templates](p. 661)

**Explanation**

It is a best practice to run containers as a non-root user user. By default, containers run as the root user user, unless the `User` directive is included in your Dockerfile. The default Linux capabilities that are assigned by Docker restrict the actions that can be run as the root user user, but only marginally. For example, a container running as the root user user does not have access to devices.

**Usage considerations**

- This control applies only to Amazon ECS task definitions that are configured with container definitions.

**Remediation for rule failure**

Set the `User` property to a non-root user user.

The examples that follow show how to implement this remediation.

**Amazon ECS Task Definition - Example**

Amazon ECS task definition configured with a container definition and a non-root user user. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ECSTaskDefinition": {
        "Type": "AWS::ECS::TaskDefinition",
        "Properties": {
            "Memory": "512",
            "ContainerDefinitions": [
                {
                    "Essential": true,
                    "Image": "nginx:latest",
                    "Name": "SampleContainerA",
                    "User": "sampleuser",
                    "Memory": 256
                }
            ]
        }
    }
}
```
YAML example

ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Memory: '512'
    ContainerDefinitions:
      - Essential: true
        Image: nginx:latest
        Name: SampleContainerA
        User: sampleuser
        Memory: 256

CT.ECS.PR.3 rule specification

```bash
# ###################################################################
## Rule Specification ##
###################################################################
#
# Rule Identifier:
#   ecs_task_definition_nonroot_user_check
#
# Description:
#   This control checks whether Amazon Elastic Container Service (ECS) task definitions run
#   as a non-root user within Amazon ECS containers.
#
# Reports on:
#   AWS::ECS::TaskDefinition
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation
#            hook document
#            And: The input document does not contain an ECS task definition resource
#            Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation
#            hook document
#            And: The input document contains an ECS task definition resource
#            And: 'ContainerDefinitions' is not present or is an empty list
#            Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation
#            hook document
#            And: The input document contains an ECS task definition resource
#            And: 'ContainerDefinitions' property is present and is not an empty list
```
# And: One or more containers defined in 'ContainerDefinitions' do not provide a 'User' property
# Then: FAIL

# Scenario: 4
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook document
# And: The input document contains an ECS task definition resource
# And: 'ContainerDefinitions' is present and is not an empty list
# And: One or more containers defined in 'ContainerDefinitions' has a 'User' property set to a root user
# Then: FAIL

# Scenario: 5
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook document
# And: The input document contains an ECS task definition resource
# And: 'ContainerDefinitions' is present and is not an empty list
# And: All containers defined in 'ContainerDefinitions' do not have a 'User' property set to a root user
# Then: PASS

# Constants
#
let ECS_TASK_DEFINITION_TYPE = "AWS::ECS::TaskDefinition"
let INPUT_DOCUMENT = this
let ROOT_USER_PATTERNS = [ 0 , "0" , "root" , /^0:.*$/ , /^root:.*$/ ]

# Assignments
#
let ecs_task_definitions = Resources.*[ Type == %ECS_TASK_DEFINITION_TYPE ]

# Primary Rules
#
rule ecs_task_definition_nonroot_user_check when is_cfn_template(%INPUT_DOCUMENT) {
  %ecs_task_definitions not empty {
    check(%ecs_task_definitions.Properties)
    <<
    [CT.ECS.PR.3]: Require any Amazon ECS task definition to specify a user that is not the root
    [FIX]: Set the 'User' property to a non-root user.
    >>
  }
}

rule ecs_task_definition_nonroot_user_check when is_cfn_hook(%INPUT_DOCUMENT, %ECS_TASK_DEFINITION_TYPE) {
  check(%INPUT_DOCUMENT.%ECS_TASK_DEFINITION_TYPE.resourceProperties)
  <<
  [CT.ECS.PR.3]: Require any Amazon ECS task definition to specify a user that is not the root
  [FIX]: Set the 'User' property to a non-root user.
  >>
}

# Parameterized Rules
#
rule check(ecs_task_definition) {
  %ecs_task_definition {
    filter_container_definitions_is_present(this)
  }[
    ContainerDefinitions[*] {
      # Scenario 3
  ]
User exists

# Scenario 4 and 5
User not in %ROOT_USER_PATTERNS
}
}

rule filter_container_definitions_is_present(ecs_task_definition) {
  %ecs_task_definition {
    # Scenario 2
    ContainerDefinitions exists
    ContainerDefinitions is_list
    ContainerDefinitions not empty
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.ECS.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Memory: '512'
    ContainerDefinitions:
      - Essential: true
        Image: nginx:latest
        Name: ExampleContainerA
        User: exampleuser
        Memory: 256
      - Name: ExampleContainerB
        Image: alpine:latest
        User: exampleuser

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
[CT.ECS.PR.4] Require Amazon ECS tasks to use 'awsvpc' networking mode

This control checks whether the networking mode for Amazon Elastic Container Service (ECS) task definitions is set to awsvpc.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ECS::TaskDefinition
- **AWS CloudFormation guard rule:** CT.ECS.PR.4 rule specification (p. 663)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ECS.PR.4 rule specification (p. 663)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.ECS.PR.4 example templates (p. 665)

Explanation

Amazon ECS recommends using the awsvpc network mode, unless you have a specific reason to use a different network mode. The awsvpc network mode simplifies container networking, and it gives you control over the ways that containerized applications communicate with each other, or with other services, within your VPCs. The awsvpc network mode provides improved security for your containers, because it empowers you to use security groups and network monitoring tools at a detailed level within your tasks. Each task has its own elastic network interface (ENI); therefore, you can include other Amazon EC2 networking features, such as VPC Flow Logs, which help you monitor traffic among your tasks.

**Usage considerations**

- This control applies only to Amazon ECS task definitions for launch types of EC2 and Fargate.

Remediation for rule failure

Set NetworkMode to awsvpc for Amazon ECS tasks that deploy to Amazon EC2 or AWS Fargate.

The examples that follow show how to implement this remediation.

**Amazon ECS Task Definition - Example**

Amazon ECS task definition configured with awsvpc networking mode. The example is shown in JSON and in YAML.
JSON example

```json
{
   "Resources": {
      "ECSTaskDefinition": {
         "Type": "AWS::ECS::TaskDefinition",
         "Properties": {
            "ContainerDefinitions": [
               {
                  "Essential": true,
                  "Image": "nginx:latest",
                  "Name": "SampleContainer"
               }
            ],
            "Memory": 512,
            "NetworkMode": "awsvpc"
         }
      }
   }
}
```

YAML example

```yaml
Resources:
  ECSTaskDefinition:
    Type: AWS::ECS::TaskDefinition
    Properties:
      ContainerDefinitions:
      - Essential: true
        Image: nginx:latest
        Name: SampleContainer
        Memory: 512
        NetworkMode: awsvpc
```

CT.ECS.PR.4 rule specification

```
# ###################################
##       Rule Specification        
####################################
#
# Rule Identifier:
#   ecs_awsvpc_networking_enabled_check
#
# Description:
#   This control checks whether the networking mode for Amazon Elastic Container Service (ECS) task definitions is set to 'awsvpc'.
# # Reports on:
# # AWS::ECS::TaskDefinition #
# # Evaluates:
# # AWS CloudFormation, AWS CloudFormation hook #
# # Rule Parameters:
# # None #
# # Scenarios:
```
# Scenario: 1
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
# And: The input document does not contain an ECS task definition resource
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
# And: The input document contains an ECS task definition resource
# And: 'RequiresCompatibilities' is present and only has one entry in the list set to
# 'EXTERNAL'
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
# And: The input document contains an ECS task definition resource
# And: 'RequiresCompatibilities' is either not present or set to a list with entries
# that include 'EC2',
# 'FARGATE' or both.
# And: 'NetworkMode' is not present
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
# And: The input document contains an ECS task definition resource
# And: 'RequiresCompatibilities' is either not present or set to a list with entries
# that include 'EC2',
# 'FARGATE' or both.
# And: 'NetworkMode' is present
# And: 'NetworkMode' is not set to 'awsvpc'
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
# hook document
# And: The input document contains an ECS task definition resource
# And: 'RequiresCompatibilities' is either not present or set to a list with entries
# that include 'EC2',
# 'FARGATE' or both.
# And: 'NetworkMode' is present
# And: 'NetworkMode' is set to 'awsvpc'
# Then: PASS

# Constants
#
let ECS_TASK_DEFINITION_TYPE = "AWS::ECS::TaskDefinition"
let ALLOWED_NETWORK_MODES = [ "awsvpc" ]
let SUPPORTED_LAUNCH_PLATFORMS = [ "EC2", "FARGATE" ]
let INPUT_DOCUMENT = this

# Assignments
#
let ecs_task_definitions = Resources.*[ Type == %ECS_TASK_DEFINITION_TYPE ]

# Primary Rules
#
rule ecs_awsvpc_networking_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%ecs_task_definitions not empty {
    check(%ecs_task_definitions.Properties)
    <<
    [CT.ECS.PR.4]: Require Amazon ECS tasks to use 'awsvpc' networking mode
    [FIX]: Set 'NetworkMode' to 'awsvpc' for Amazon ECS tasks that deploy to Amazon EC2 or AWS Fargate.
    >>
}
Proactive controls

rule ecs_awsvpc_networking_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ECS_TASK_DEFINITION_TYPE) {
    check(%INPUT_DOCUMENT.%ECS_TASK_DEFINITION_TYPE.resourceProperties)
    <<
    [CT.ECS.PR.4]: Require Amazon ECS tasks to use 'awsvpc' networking mode
    [FIX]: Set 'NetworkMode' to 'awsvpc' for Amazon ECS tasks that deploy to Amazon EC2 or AWS Fargate.
    >>
}

# Parameterized Rules
#
rule check(ecs_task_definition) {
    %ecs_task_definition [ filter_external_task_definitions(this) ] {
        # Scenario 3
        NetworkMode exists
        # Scenario 4 and 5
        NetworkMode is_string
        NetworkMode in %ALLOWED_NETWORK_MODES
    }
}

rule filter_external_task_definitions(ecs_task_definition) {
    %ecs_task_definition {
        # Scenario 2
        RequiresCompatibilities not exists or
        filter_supported_task_definitions(RequiresCompatibilities)
    }
}

rule filter_supported_task_definitions(requires_compatibilities) {
    %requires_compatibilities {
        this is_list
        this not empty
        some this[*] in %SUPPORTED_LAUNCH_PLATFORMS
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.ECS.PR.4 example templates
You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    ContainerDefinitions:
    - Essential: true
      Image: nginx:latest
      Name: ExampleContainer
      Memory: 512
      NetworkMode: awsvpc

[CT.ECS.PR.5] Require an active Amazon ECS task definition to have a logging configuration

This control checks whether Amazon Elastic Container Service (ECS) task definitions have a logging configuration specified.

- **Control objective**: Establish logging and monitoring
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::ECS::TaskDefinition
- **AWS CloudFormation guard rule**: [CT.ECS.PR.5 rule specification](p. 668)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ECS.PR.5 rule specification](p. 668)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ECS.PR.5 example templates](p. 670)

Explanation

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon Elastic Container Service (ECS) and your AWS environments. We recommend that you collect monitoring data from all parts of your AWS environment, because this data can help you debug a multi-point failure, if such a failure occurs.

**Usage considerations**

- This control applies only to Amazon ECS task definitions that are configured with container definitions.
Remediation for rule failure

For each container definition, within LogConfiguration set the LogDriver property to a supported log driver.

The examples that follow show how to implement this remediation.

Amazon ECS Task Definition - Example

Amazon ECS task definition configured to send log information to Amazon CloudWatch Logs for each container definition. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "ECSTaskDefinition": {
      "Type": "AWS::ECS::TaskDefinition",
      "Properties": {
         "Memory": "512",
         "ContainerDefinitions": [
            {
               "Name": "ContainerA",
               "Image": "nginx:latest",
               "Essential": true,
               "LogConfiguration": {
                  "LogDriver": "awslogs",
                  "Options": {
                     "awslogs-group": {
                        "Ref": "LogGroup"
                     },
                     "awslogs-region": {
                        "Ref": "AWS::Region"
                     }
                  }
               }
            },
            {
               "Name": "ContainerB",
               "Image": "nginx:latest",
               "LogConfiguration": {
                  "LogDriver": "awslogs",
                  "Options": {
                     "awslogs-group": {
                        "Ref": "LogGroup"
                     },
                     "awslogs-stream-prefix": "Container-B-LogStream",
                     "awslogs-region": {
                        "Ref": "AWS::Region"
                     }
                  }
               }
            }
         ]
      }
   }
}
```

**YAML example**

```yaml
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Memory: 512
    ContainerDefinitions:
      - Name: ContainerA
        Image: nginx:latest
        Essential: true
        LogConfiguration:
          LogDriver: awslogs
          Options:
            awslogs-group: !Ref LogGroup
            awslogs-stream-prefix: Container-A-LogStream
            awslogs-region: !Ref AWS::Region
      - Name: ContainerB
        Image: nginx:latest
        LogConfiguration:
          LogDriver: awslogs
          Options:
            awslogs-group: !Ref LogGroup
            awslogs-stream-prefix: Container-B-LogStream
            awslogs-region: !Ref AWS::Region
```
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Memory: '512'
    ContainerDefinitions:
      - Name: ContainerA
        Image: nginx:latest
        Essential: true
        LogConfiguration:
          LogDriver: awslogs
          Options:
            awslogs-group: !Ref 'LogGroup'
            awslogs-stream-prefix: Container-A-LogStream
            awslogs-region: !Ref 'AWS::Region'
      - Name: ContainerB
        Image: nginx:latest
        LogConfiguration:
          LogDriver: awslogs
          Options:
            awslogs-group: !Ref 'LogGroup'
            awslogs-stream-prefix: Container-B-LogStream
            awslogs-region: !Ref 'AWS::Region'

CT.ECS.PR.5 rule specification

# ###################################################################
##       Rule Specification        ##
###################################################################
#
# Rule Identifier:
#   ecs_task_definition_log_configuration_check
#
# Description:
#   This control checks whether Amazon Elastic Container Service (ECS) task definitions
#   have a logging configuration specified.
#
# Reports on:
#   AWS::ECS::TaskDefinition
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation
#     hook document
#     And: The input document does not contain an ECS task definition resource
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation
#     hook document
#     And: The input document contains an ECS task definition resource
#     And: 'ContainerDefinitions' property is not present or is an empty list
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation
#     hook document
#     And: The input document contains an ECS task definition resource
# And: 'ContainerDefinitions' property is present and is not an empty list
# And: One or more containers defined in 'ContainerDefinitions' do not have
'LogConfiguration' set or it is set
to an empty struct
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
hook document
# And: The input document contains an ECS task definition resource
# And: 'ContainerDefinitions' property is present and is not an empty list
# And: One or more containers defined in 'ContainerDefinitions' have
'LogConfiguration' property present
# And: 'LogConfiguration.LogDriver' is not present or is set to an empty string
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation
hook document
# And: The input document contains an ECS task definition resource
# And: 'ContainerDefinitions' property is present
# And: All containers defined in 'ContainerDefinitions' have 'LogConfiguration'
property present
# And: 'LogConfiguration.LogDriver' is present and set to a non-empty string
# Then: PASS

# Constants
#
let ECS_TASK_DEFINITION_TYPE = "AWS::ECS::TaskDefinition"
let INPUT_DOCUMENT = this
#
# Assignments
#
let ecs_task_definitions = Resources.*[ Type == %ECS_TASK_DEFINITION_TYPE ]
#
# Primary Rules
#
rule ecs_task_definition_log_configuration_check when is_cfn_template(%INPUT_DOCUMENT)
%ecs_task_definitions not empty { 
  check(%ecs_task_definitions.Properties) 
  \[CT.ECS.PR.5\]: Require an active Amazon ECS task definition to have a logging
configuration
  \[FIX\]: For each container definition, within 'LogConfiguration' set the
'LogDriver' property to a supported log driver.
} 

rule ecs_task_definition_log_configuration_check when is_cfn_hook(%INPUT_DOCUMENT,
%ECS_TASK_DEFINITION_TYPE) { 
  check(%INPUT_DOCUMENT.%ECS_TASK_DEFINITION_TYPE.resourceProperties) 
  \[CT.ECS.PR.5\]: Require an active Amazon ECS task definition to have a logging
configuration
  \[FIX\]: For each container definition, within 'LogConfiguration' set the
'LogDriver' property to a supported log driver.
} 

# Parameterized Rules
#
rule check(ecs_task_definition) { 
  %ecs_task_definition [ 
    filter_container_definitions_is_present(this) 
  ] 
}
Proactive controls

CT.ECS.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
LogGroup:
  Type: AWS::Logs::LogGroup
  DeletionPolicy: Delete
  UpdateReplacePolicy: Delete
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Memory: '512'
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
LogGroup:
  Type: AWS::Logs::LogGroup
  DeletionPolicy: Delete
  UpdateReplacePolicy: Delete
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Memory: '512'
    ContainerDefinitions:
      - Name: ContainerA
        Essential: true
        Image: nginx:latest
      - Name: ContainerB
        Image: nginx:latest
        LogConfiguration:
          LogDriver: awslogs
          Options:
            awslogs-group:
              Ref: LogGroup
            awslogs-stream-prefix: Container-B-LogStream
            awslogs-region:
              Ref: AWS::Region

[CT.ECS.PR.6] Require Amazon ECS containers to allow read-only access to the root filesystem

This control checks whether Amazon Elastic Container Service (Amazon ECS) task definitions have been configured to require read-only access to container root filesystems.

• **Control objective:** Enforce least privilege
• **Implementation:** AWS CloudFormation Guard Rule
• Control behavior: Proactive
• Resource types: AWS::ECS::TaskDefinition
• AWS CloudFormation guard rule: CT.ECS.PR.6 rule specification (p. 673)

Details and examples

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ECS.PR.6 rule specification (p. 673)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.ECS.PR.6 example templates (p. 675)

Explanation

Enabling this option reduces security attack vectors. When it is enabled, the container instance’s filesystem cannot be tampered with or written to unless it has explicitly granted read-write permissions on its filesystem folder and directories. This control adheres to the principle of least privilege.

Usage considerations

• This control is incompatible with Amazon ECS task definitions that use Windows containers.

Remediation for rule failure

Set the ReadonlyRootFilesystem property to true for all ContainerDefinitions.

The examples that follow show how to implement this remediation.

Amazon ECS Task Definition - Example

Amazon ECS task definition with read-only access to container root filesystems. The example is shown in JSON and in YAML.

JSON example

```json
{
    "ECSTaskDefinition": {
        "Type": "AWS::ECS::TaskDefinition",
        "Properties": {
            "ContainerDefinitions": [ {
                "Essential": true,
                "Image": "nginx:latest",
                "Name": "SampleContainerA",
                "ReadonlyRootFilesystem": true
            }, {
                "Image": "alpine:latest",
                "Name": "SampleContainerB",
                "ReadonlyRootFilesystem": true
            } ],
            "Memory": "512"
        }
    }
}
```

YAML example

```yaml
672
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    ContainerDefinitions:
    - Essential: true
      Image: nginx:latest
      Name: SampleContainerA
      ReadonlyRootFilesystem: true
    - Image: alpine:latest
      Name: SampleContainerB
      ReadonlyRootFilesystem: true
      Memory: '512'

CT.ECS.PR.6 rule specification

# ###############################################################################
##       Rule Specification        ##
# ###############################################################################
# Rule Identifier:
#   ecs_containers_readonly_access_check
#
# Description:
#   This control checks whether Amazon Elastic Container Service (Amazon ECS) task
# definitions have been configured to require read-only access to container root
# filesystems.
#
# Reports on:
#   AWS::ECS::TaskDefinition
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook
document
#   And: The input document does not contain an Amazon ECS task definition resource
#   Then: SKIP
# Scenario: 2
#   Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook
document
#   And: The input document contains an Amazon ECS task definition resource
#   And: 'ContainerDefinitions' property is not present or is empty
#   Then: SKIP
# Scenario: 3
#   Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook
document
#   And: The input document contains an Amazon ECS task definition resource
#   And: 'ContainerDefinitions' property is present
#   And: One or more containers defined in 'ContainerDefinitions' do not have
#        'ReadonlyRootFilesystem' present
#   Then: FAIL
# Scenario: 4
#   Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook
document
#   And: The input document contains an Amazon ECS task definition resource
#   And: 'ContainerDefinitions' property is present
And: One or more containers defined in 'ContainerDefinitions' have the value of 'ReadonlyRootFilesystem' set to bool(false)
Then: FAIL
Scenario: 5
Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook document
And: The input document contains an Amazon ECS task definition resource
And: 'ContainerDefinitions' property is present
And: All containers defined in 'ContainerDefinitions' have the value of 'ReadonlyRootFilesystem' set to bool(true)
Then: PASS

# Constants
let ECS_TASK_DEFINITION_TYPE = "AWS::ECS::TaskDefinition"
let INPUT_DOCUMENT = this

# Assignments
let ecs_task_definitions = Resources.*[ Type == %ECS_TASK_DEFINITION_TYPE ]

# Primary Rules
# rule ecs_containers_readonly_access_check when is_cfn_template(%INPUT_DOCUMENT)
%ecs_task_definitions not empty {
  check(%ecs_task_definitions.Properties)
<<<
  [CT.ECS.PR.6]: Require Amazon ECS containers to allow read-only access to the root filesystem
  [FIX]: Set the 'ReadonlyRootFilesystem' property to 'true' for all 'ContainerDefinitions'.
  >>>
}

rule ecs_containers_readonly_access_check when is_cfn_hook(%INPUT_DOCUMENT, %ECS_TASK_DEFINITION_TYPE) {
  check(%INPUT_DOCUMENT.%ECS_TASK_DEFINITION_TYPE.resourceProperties)
<<<
  [CT.ECS.PR.6]: Require Amazon ECS containers to allow read-only access to the root filesystem
  [FIX]: Set the 'ReadonlyRootFilesystem' property to 'true' for all 'ContainerDefinitions'.
  >>>
}

# Parameterized Rules
# rule check(ecs_task_definition) {
%ecs_task_definition [ filter_container_definitions_is_present(this) ]{
  ContainerDefinitions[*] {
    # Scenario 3
    ReadonlyRootFilesystem exists
    # Scenario 4
    ReadonlyRootFilesystem == true
  }
}
rule filter_container_definitions_is_present(ecs_task_definition) {

%ecs_task_definition {
    # Scenario 2
    ContainerDefinitions exists
    ContainerDefinitions is_list
    ContainerDefinitions not empty
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.ECS.PR.6 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
ECSTaskDefinition:
    Type: AWS::ECS::TaskDefinition
    Properties:
        ContainerDefinitions:
        - Essential: true
        Image: nginx:latest
        Name: SampleContainerA
        ReadonlyRootFilesystem: true
        - Image: alpine:latest
        Name: SampleContainerB
        ReadonlyRootFilesystem: true
        Memory: '512'

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ECSTaskDefinition:
    Type: AWS::ECS::TaskDefinition
    Properties:
        ContainerDefinitions:
        - Essential: true
        Image: nginx:latest
        Name: SampleContainerA
        ReadonlyRootFilesystem: false
        - Image: alpine:latest
        Name: SampleContainerB
        ReadonlyRootFilesystem: false
        Memory: '512'
[CT.ECS.PR.7] Require an Amazon ECS task definition to have a specific memory usage limit

This control checks whether Amazon Elastic Container Service (ECS) task definitions have specified a memory limit for container definitions.

- **Control objective:** Improve availability
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ECS::TaskDefinition
- **AWS CloudFormation guard rule:** [CT.ECS.PR.7 rule specification (p. 677)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: [CT.ECS.PR.7 rule specification (p. 677)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ECS.PR.7 example templates (p. 679)]

**Explanation**

We recommend that you specify the maximum memory available to your containers, because this limit protects your resources in the event of malicious access to your containers.

**Usage considerations**

- This control applies only to Amazon ECS task definitions that are configured with container definitions.

**Remediation for rule failure**

Set the Memory property in ContainerDefinitions for Amazon ECS task definitions.

The examples that follow show how to implement this remediation.

**Amazon ECS Task Definition - Example**

Amazon ECS task definition configured with a specified memory limit for container definitions. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ECSTaskDefinition": {
    "Type": "AWS::ECS::TaskDefinition",
    "Properties": {
      "ContainerDefinitions": [
        {
          "Essential": true,
          "Image": "nginx:latest",
          "Name": "SampleContainer",
          "Memory": 256
        }
      ]
    }
  }
}
```
YAML example

ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    ContainerDefinitions:
      - Essential: true
        Image: nginx:latest
        Name: SampleContainer
        Memory: 256

CT.ECS.PR.7 rule specification

# #############################################################################
##       Rule Specification        ##
# #############################################################################

# Rule Identifier:
#   ecs_task_definition_memory_hard_limit_check
#
# Description:
#   This control checks whether Amazon Elastic Container Service (ECS) task definitions
#   have specified a memory limit for container definitions.
#
# Reports on:
#   AWS::ECS::TaskDefinition
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation
#           hook document
#     And: The input document does not contain an ECS task definition resource
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook
#            document
#     And: The input document contains an ECS task definition resource
#     And: 'ContainerDefinitions' property is not present or is an empty list
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook
#            document
#     And: The input document contains an ECS task definition resource
#     And: 'ContainerDefinitions' property is present and is not an empty list
#     And: One or more containers defined in 'ContainerDefinitions' do not have 'Memory'
#          property set
#     Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook
document  
And: The input document contains an ECS task definition resource  
And: 'ContainerDefinitions' property is present and is not an empty list  
And: One or more containers defined in 'ContainerDefinitions' have 'Memory' property set to an integer
  value less than four (< 4)  
Then: FAIL  
# Scenario: 5  
# Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook
document  
And: The input document contains an ECS task definition resource  
And: 'ContainerDefinitions' property is present  
And: All containers defined in 'ContainerDefinitions' have 'Memory' property set to an integer value
  greater than or equal to four (>= 4)  
Then: PASS

# Constants
let ECS_TASK_DEFINITION_TYPE = "AWS::ECS::TaskDefinition"
let INPUT_DOCUMENT = this

# Assignments
let ecs_task_definitions = Resources.*[ Type == %ECS_TASK_DEFINITION_TYPE ]

# Primary Rules
rule ecs_task_definition_memory_hard_limit_check when is_cfn_template(%INPUT_DOCUMENT)  
%ecs_task_definitions not empty {  
  check(%ecs_task_definitions.Properties)  
  <<
  [CT.ECS.PR.7]: Require an Amazon ECS task definition to have a specific memory usage limit
  [FIX]: Set the 'Memory' property in 'ContainerDefinitions' for Amazon ECS task definitions.
  >>
}

rule ecs_task_definition_memory_hard_limit_check when is_cfn_hook(%INPUT_DOCUMENT,  
%ECS_TASK_DEFINITION_TYPE) {  
  check(%INPUT_DOCUMENT.%ECS_TASK_DEFINITION_TYPE.resourceProperties)  
  <<
  [CT.ECS.PR.7]: Require an Amazon ECS task definition to have a specific memory usage limit
  [FIX]: Set the 'Memory' property in 'ContainerDefinitions' for Amazon ECS task definitions.
  >>
}

# Parameterized Rules
# rule check(ecs_task_definition) {
#  %ecs_task_definition [filter_container_definitions_is_present(this)][
# ContainerDefinitions[*] {
#    # Scenario 3  
#    Memory exists

# Scenario 4 and 5
Memory >= 4
rule filter_container_definitions_is_present(ecs_task_definition) {
    %ecs_task_definition {
        # Scenario 2
        ContainerDefinitions exists
        ContainerDefinitions is_list
        ContainerDefinitions not empty
    }
}

# Utility Rules
# is_cfn_template(doc) {
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.ECS.PR.7 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
ECSTaskDefinition:
    Type: AWS::ECS::TaskDefinition
    Properties:
        ContainerDefinitions:
            - Essential: true
              Image: nginx:latest
              Name: ExampleContainerA
              Memory: 256
            - Image: alpine:latest
              Name: ExampleContainerB
              Memory: 512

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ECSTaskDefinition:
    Type: AWS::ECS::TaskDefinition
    Properties:
        Memory: "512"
        ContainerDefinitions:
            - Essential: true
              Image: nginx:latest
              Name: ExampleContainerA
[CT.ECS.PR.8] Require Amazon ECS task definitions to have secure networking modes and user definitions

This control checks whether Amazon Elastic Container Service (ECS) task definitions that use host networking mode have a privileged container definition, and whether they specify a non-root user definition.

- **Control objective:** Manage vulnerabilities
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ECS::TaskDefinition
- **AWS CloudFormation guard rule:** [CT.ECS.PR.8 rule specification](p. 682)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ECS.PR.8 rule specification](p. 682)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ECS.PR.8 example templates](p. 685)

Explanation

If a task definition has elevated privileges, it implies that a customer has specifically chosen that configuration. This control checks for unexpected privilege escalation, which occurs when a task definition enables host networking, but a customer has not opted into elevated privileges.

**Usage considerations**

- This control applies only to Amazon ECS task definitions that include host networking mode and one or more container definitions.
- This control is incompatible with Amazon ECS task definitions that use Windows containers.

Remediation for rule failure

For Amazon ECS task definitions that use host networking mode, your container definitions must set the `User` property to a non-root user. Also, to opt into elevated privileges, configure containers to run in privileged mode by setting the `Privileged` property to `true`.

The examples that follow show how to implement this remediation.

Amazon ECS Task Definition - Example One

Amazon ECS task definition with host networking mode configured for privileged container definitions. The example is shown in JSON and in YAML.

**JSON example**
YAML example

ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Memory: '512'
    NetworkMode: host
  ContainerDefinitions:
    - Name: SampleContainerA
      User: root
      Privileged: true
      Image: nginx:latest
      Essential: true
    - Name: SampleContainerB
      User: root
      Privileged: true
      Image: alpine:latest

The examples that follow show how to implement this remediation.

Amazon ECS Task Definition - Example Two

Amazon ECS task definition with host networking mode configured for non-root user container definitions and privileged mode deactivated, by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.

JSON example

```json
{
  "ECSTaskDefinition": {
    "Type": "AWS::ECS::TaskDefinition",
    "Properties": {
      "Memory": "512",
      "NetworkMode": "host",
      "ContainerDefinitions": [
        {
          "Name": "SampleContainerA",
          "User": "root",
          "Privileged": true,
          "Image": "nginx:latest",
          "Essential": true
        },
        {
          "Name": "SampleContainerB",
          "User": "root",
          "Privileged": true,
          "Image": "alpine:latest"
        }
      ]
    }
  }
}
```
"NetworkMode": "host",
"ContainerDefinitions": [
{
   "Name": "SampleContainerA",
   "User": "root",
   "Privileged": true,
   "Image": "nginx:latest",
   "Essential": true
},
{
   "Name": "SampleContainerB",
   "User": "root",
   "Privileged": true,
   "Image": "alpine:latest"
}
]

YAML example

ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Memory: '512'
    NetworkMode: host
    ContainerDefinitions:
    - Name: SampleContainerA
      User: root
      Privileged: true
      Image: nginx:latest
      Essential: true
    - Name: SampleContainerB
      User: root
      Privileged: true
      Image: alpine:latest

CT.ECS.PR.8 rule specification

# #############################################################################
##       Rule Specification       #
# #############################################################################
#
# Rule Identifier:  
#   ecs_task_definition_user_for_host_mode_check
#
# Description: 
#   This control checks whether Amazon Elastic Container Service (ECS) task definitions 
#   that use 'host' networking mode have a privileged container definition, and whether they 
#   specify a non-root user definition.
#
# Reports on: 
#   AWS::ECS::TaskDefinition
#
# Evaluates: 
#   AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain an ECS task definition resource
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ECS task definition resource
# And: 'ContainerDefinitions' property is not present or is an empty list
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ECS task definition resource
# And: 'ContainerDefinitions' property is present and is not an empty list
# And: 'NetworkMode' property is either not present or set to a value other than 'host'
# Then: SKIP
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ECS task definition resource
# And: 'ContainerDefinitions' property is present and is not an empty list
# And: 'NetworkMode' property is present and set to 'host'
# And: A container defined in 'ContainerDefinitions' has 'Privileged' property not set or is set as bool(false)
# And: This same container either does not have the 'User' property set or has it set to a value that translates to root user
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ECS task definition resource
# And: 'ContainerDefinitions' property is present
# And: 'NetworkMode' property is present and set to 'host'
# And: All Containers defined in 'ContainerDefinitions' either have the 'Privileged' property set to bool(true)
# or have their 'User' property set to a value that does not translate to root user
# Then: PASS
#
# Constants
#
let ECS_TASK_DEFINITION_TYPE = "AWS::ECS::TaskDefinition"
let INPUT_DOCUMENT = this
let ROOT_USER_PATTERNS = [ 0 , "0" , "root" , /^0:.*$/ , /^root:.*$/ ]
let VALID_NETWORK_MODES = [ "host" ]
#
# Assignments
#
let ecs_task_definitions = Resources.*[ Type == %ECS_TASK_DEFINITION_TYPE ]
#
# Primary Rules
#
rule ecs_task_definition_user_for_host_mode_check when is_cfn_template(%INPUT_DOCUMENT) %ecs_task_definitions not empty {
    check(%ecs_task_definitions.Properties)
Proactive controls

[CT.ECS.PR.8]: Require Amazon ECS task definitions to have secure networking modes and user definitions

[FIX]: For Amazon ECS task definitions that use 'host' networking mode, your container definitions must set the 'User' property to a non-root user. Also, to opt into elevated privileges, configure containers to run in privileged mode by setting the 'Privileged' property to 'true'.

rule ecs_task_definition_user_for_host_mode_check when is_cfn_hook(%INPUT_DOCUMENT, %ECS_TASK_DEFINITION_TYPE) {
  check(%INPUT_DOCUMENT.%ECS_TASK_DEFINITION_TYPE.resourceProperties)
}

rule check(ecs_task_definition) {
  %ecs_task_definition [
    filter_nw_mode_container_definitions(this)
  ]{
    ContainerDefinitions[*] {
      # Scenario 4 and 5
      check_elevated_privilege_containers(this) or
      check_nonroot_user_containers(this)
    }
  }
}

rule check_elevated_privilege_containers(container_definition) {
  %container_definition {
    Privileged exists
    Privileged == true
  }
}

rule check_nonroot_user_containers(container_definition) {
  %container_definition {
    User exists
    User not in %ROOT_USER_PATTERNS
  }
}

rule filter_nw_mode_container_definitions(ecs_task_definition) {
  %ecs_task_definition {
    # Scenario 2
    ContainerDefinitions exists
    ContainerDefinitions is_list
    ContainerDefinitions not empty
    # Scenario 3
    NetworkMode exists
    NetworkMode is_string
    NetworkMode in %VALID_NETWORK_MODES
  }
}
# Utility Rules

```haskell
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
	rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

## CT.ECS.PR.8 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```json
Resources:
  ECSTaskDefinition:
    Type: AWS::ECS::TaskDefinition
    Properties:
      Memory: '512'
      NetworkMode: host
      ContainerDefinitions:
        - Name: ExampleContainerA
          User: root
          Privileged: true
          Image: nginx:latest
          Essential: true
        - Name: ExampleContainerB
          User: root
          Privileged: true
          Image: alpine:latest
```

**FAIL Example** - Use this template to verify that the control prevents non-compliant resource creation.

```json
Resources:
  ECSTaskDefinition:
    Type: AWS::ECS::TaskDefinition
    Properties:
      Memory: '512'
      NetworkMode: host
      ContainerDefinitions:
        - Name: ExampleContainerA
          User: root
          Privileged: true
          Image: nginx:latest
          Essential: true
        - Name: ExampleContainerB
          Image: alpine:latest
          User: root
```

685
[CT.ECS.PR.9] Require Amazon ECS services not to assign public IP addresses automatically

This control checks whether your Amazon Elastic Container Service (Amazon ECS) service resources are configured to assign public IP addresses automatically.

- **Control objective:** Limit network access, Enforce least privilege
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ECS::Service
- **AWS CloudFormation guard rule:** CT.ECS.PR.9 rule specification (p. 688)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ECS.PR.9 rule specification (p. 688)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.ECS.PR.9 example templates (p. 690)

**Explanation**

A public IP address is an IP address that is reachable from the internet. If you launch your Amazon ECS instances with a public IP address, then your Amazon ECS instances are reachable from the internet. Amazon ECS services should not be publicly accessible, because it may allow unintended access to your container application servers.

**Remediation for rule failure**

Set AssignPublicIp in NetworkConfiguration.AwsvpcConfiguration to DISABLED.

The examples that follow show how to implement this remediation.

**Amazon ECS Service - Example One**

Amazon ECS service configured to disallow automatic public IP address assignment, by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ECSService": {
        "Type": "AWS::ECS::Service",
        "Properties": {
            "Cluster": {
                "Ref": "ECSCluster"
            },
            "DesiredCount": 0,
            "TaskDefinition": {
                "Ref": "ECSTaskDefinition"
            },
            "LaunchType": "FARGATE",
            "NetworkConfiguration": {
                "AwsvpcConfiguration": {
                    "Subnets": [
                        {
                            "Ref": "SubnetOne"
                        }
                    ]
                }
            }
        }
    }
}
```
YAML example

ECSService:
  Type: AWS::ECS::Service
  Properties:
    Cluster: !Ref 'ECSCluster'
    DesiredCount: 0
    TaskDefinition: !Ref 'ECSTaskDefinition'
    LaunchType: FARGATE
    NetworkConfiguration:
      AwsvpcConfiguration:
        Subnets:
        - !Ref 'SubnetOne'
        - !Ref 'SubnetTwo'

The examples that follow show how to implement this remediation.

Amazon ECS Service - Example Two

Amazon ECS service configured to disallow automatic public IP address assignment, by means of the AssignPublicIp property. The example is shown in JSON and in YAML.

JSON example

```json

{
  "ECSService": {
    "Type": "AWS::ECS::Service",
    "Properties": {
      "Cluster": {
        "Ref": "ECSCluster"
      },
      "DesiredCount": 0,
      "TaskDefinition": {
        "Ref": "ECSTaskDefinition"
      },
      "LaunchType": "FARGATE",
      "NetworkConfiguration": {
        "AwsvpcConfiguration": {
          "AssignPublicIp": "DISABLED",
          "Subnets": [
            {
              "Ref": "SubnetOne"
            },
            {
              "Ref": "SubnetTwo"
            }
          ]
        }
      }
    }
  }
}
```
YAML example

ECSService:
  Type: AWS::ECS::Service
  Properties:
    Cluster: !Ref 'ECSCluster'
    DesiredCount: 0
    TaskDefinition: !Ref 'ECSTaskDefinition'
    LaunchType: FARGATE
    NetworkConfiguration:
      AwsvpcConfiguration:
        AssignPublicIp: DISABLED
        Subnets:
          - !Ref 'SubnetOne'
          - !Ref 'SubnetTwo'

CT.ECS.PR.9 rule specification

# ###################################################################
#                  Rule Specification                               
# ###################################################################
#
# Rule Identifier:
#   ecs_service_assign_public_ip_disabled_check
#
# Description:
#   This control checks whether your Amazon Elastic Container Service (Amazon ECS) service resources are configured to assign public IP addresses automatically.
#
# Reports on:
#   AWS::ECS::Service
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain an Amazon ECS service resource
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Amazon ECS service resource
#     And: 'NetworkConfiguration' property is not present
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Amazon ECS service resource

688
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Amazon ECS service resource
# And: 'NetworkConfiguration.AwsvpcConfiguration' property is present
# And: 'AssignPublicIp' property is present and set to 'ENABLED'
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Amazon ECS service resource
# And: 'NetworkConfiguration.AwsvpcConfiguration' property is present
# And: 'AssignPublicIp' property is not present
# Then: PASS
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Amazon ECS service resource
# And: 'NetworkConfiguration.AwsvpcConfiguration' property is present
# And: 'AssignPublicIp' property is present and set to 'DISABLED'
# Then: PASS

# Constants
#
let ECS_SERVICE_TYPE = "AWS::ECS::Service"
let INPUT_DOCUMENT = this
#
# Assignments
#
let ecs_services = Resources.*[ Type == %ECS_SERVICE_TYPE ]
#
# Primary Rules
#
rule ecs_service_assign_public_ip_disabled_check when is_cfn_template(%INPUT_DOCUMENT) {
  %ecs_services not empty {
    check(%ecs_services.Properties)
    <-
    [CT.ECS.PR.9]: Require Amazon ECS services not to assign public IP addresses automatically
    [FIX]: Set 'AssignPublicIp' in 'NetworkConfiguration.AwsvpcConfiguration' to 'DISABLED'.
    >>
  }
}

rule ecs_service_assign_public_ip_disabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ECS_SERVICE_TYPE) {
  check(%INPUT_DOCUMENT.%ECS_SERVICE_TYPE.resourceProperties)
  <-
  [CT.ECS.PR.9]: Require Amazon ECS services not to assign public IP addresses automatically
  [FIX]: Set 'AssignPublicIp' in 'NetworkConfiguration.AwsvpcConfiguration' to 'DISABLED'.
  >>
}
#
# Parameterized Rules
#
rule check(ecs_service) {
  %ecs_service [filter_ecs_service_with_vpc_configuration(this)] {
    NetworkConfiguration {
CT.ECS.PR.9 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
[CT.ECS.PR.10] Require that Amazon ECS task definitions do not share the host's process namespace

This control checks whether Amazon Elastic Container Service (ECS) task definitions are configured to share a host's process namespace with its containers.

- **Control objective:** Protect configurations, Enforce least privilege
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ECS::TaskDefinition
• **AWS CloudFormation guard rule**: [CT.ECS.PR.10 rule specification (p. 694)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: [CT.ECS.PR.10 rule specification (p. 694)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ECS.PR.10 example templates (p. 696)]

**Explanation**

A process ID (PID) namespace provides separation between processes. It prevents system processes from being visible to other processes, and it allows PIDs to be reused, including PID 1. If the host’s PID namespace is shared with containers, those containers can see all of the processes on the host system. Process visibility reduces the benefit of process-level isolation between the host and the containers. Reduced isolation can allow unauthorized access to processes on the host itself, including the ability to manipulate and terminate the host’s processes. As a best practice, do not share the host’s process namespace with containers running on the host.

**Usage considerations**

- This control applies only to Amazon ECS task definitions that are configured with container definitions.
- This control is not compatible with Amazon ECS task definitions that are configured to run on AWS Fargate, or definitions that use Windows containers.

**Remediation for rule failure**

Omit the PidMode property, or set PidMode to task.

The examples that follow show how to implement this remediation.

**Amazon ECS Task Definition - Example One**

Amazon ECS task definition configured with a task-level process namespace, by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "TaskDefinition": {
        "Type": "AWS::ECS::TaskDefinition",
        "Properties": {
            "Memory": "512",
            "ContainerDefinitions": [
                {
                    "Essential": true,
                    "Image": "nginx:latest",
                    "Name": "SampleContainer"
                }
            ]
        }
    }
}
```
The examples that follow show how to implement this remediation.

Amazon ECS Task Definition - Example Two

Amazon ECS task definition configured with a task-level process namespace, by means of the PidMode property. The example is shown in JSON and in YAML.

JSON example

```
{
  "TaskDefinition": {
    "Type": "AWS::ECS::TaskDefinition",
    "Properties": {
      "Memory": "512",
      "ContainerDefinitions": [
        {
          "Essential": true,
          "Image": "nginx:latest",
          "Name": "SampleContainer"
        }
      ],
      "PidMode": "task"
    }
  }
}
```

YAML example

```
TaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Memory: '512'
    ContainerDefinitions:
      - Essential: true
        Image: nginx:latest
        Name: SampleContainer
    PidMode: task
```

CT.ECS.PR.10 rule specification

```
# ###################################
##       Rule Specification        ##
```
# Rule Identifier:
#   ecs_task_definition_pid_mode_check

# Description:
#   This control checks whether Amazon Elastic Container Service (ECS) task definitions are
#   configured to share a host's process namespace with its containers.

# Reports on:
#   AWS::ECS::TaskDefinition

# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
#   None

# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#           And: The input document does not contain an ECS task definition resource
#           Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#           And: The input document contains an ECS task definition resource
#           And: 'PidMode' is provided as an empty string
#           Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#           And: The input document contains an ECS task definition resource
#           And: 'PidMode' is set to 'host'
#           Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#           And: The input document contains an ECS task definition resource
#           And: 'PidMode' is not present
#           Then: PASS
#   Scenario: 5
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#           And: The input document contains an ECS task definition resource
#           And: 'PidMode' is provided as a non-empty string that is not 'host'
#           Then: PASS

# Constants
#
let ECS_TASK_DEFINITION_TYPE = "AWS::ECS::TaskDefinition"
let INPUT_DOCUMENT = this

# Assignments
#
let ecs_task_definitions = Resources.*[ Type == %ECS_TASK_DEFINITION_TYPE ]

# Primary Rules
#
rule ecs_task_definition_pid_mode_check when is_cfn_template(%INPUT_DOCUMENT)
  %ecs_task_definitions not empty {
    check(%ecs_task_definitions.Properties)
  }
[CT.ECS.PR.10]: Require that Amazon ECS task definitions do not share the host's process namespace
   [FIX]: Omit the 'PidMode' property, or set 'PidMode' to 'task'.

rule ecs_task_definition_pid_mode_check when is_cfn_hook(%INPUT_DOCUMENT, %ECS_TASK_DEFINITION_TYPE) {
    check(%INPUT_DOCUMENT.%ECS_TASK_DEFINITION_TYPE.resourceProperties)
    [CT.ECS.PR.10]: Require that Amazon ECS task definitions do not share the host's process namespace
    [FIX]: Omit the 'PidMode' property, or set 'PidMode' to 'task'.

} # Parameterized Rules
# rule check(ecs_task_definition) {
%ecs_task_definition {
    # Scenario 2
    PidMode not exists or
    # Scenario 3 and 4
    check_pidmode_value(PidMode)
}

rule check_pidmode_value(pid_mode) {
    %pid_mode {
        check_is_string_and_not_empty(this)
        this != "host"
    }
}

# Utility Rules
# rule is_cfn_template(doc) {
%doc {
    AWSTemplateFormatVersion exists or
    Resources exists
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

rule check_is_string_and_not_empty(value) {
    %value {
        this is_string
        this != /\s*\z/
    }
}

CT.ECS.PR.10 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.
Resources:

TaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Memory: '512'
    ContainerDefinitions:
      - Essential: true
        Image: nginx:latest
        Name: ExampleContainer
        PidMode: task

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:

TaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Memory: '512'
    ContainerDefinitions:
      - Essential: true
        Image: nginx:latest
        Name: ExampleContainer
        PidMode: host

[CT.ECS.PR.11] Require an Amazon ECS container to run as non-privileged

This control checks whether container definitions in Amazon Elastic Container Service (ECS) task definitions are configured with elevated privileges.

- **Control objective:** Enforce least privilege
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ECS::TaskDefinition
- **AWS CloudFormation guard rule:** CT.ECS.PR.11 rule specification (p. 698)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ECS.PR.11 rule specification (p. 698)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ECS.PR.11 example templates (p. 700)]

Explanation

We recommend that you remove elevated privileges from your Amazon ECS task definitions. When the privilege parameter is `true`, the container can operate with elevated privileges on the host container instance. These privileges are similar to the root user privileges.

Usage considerations

- This control applies only to Amazon ECS task definitions that are configured with container definitions.
• This control is incompatible with Amazon ECS task definitions that use Windows containers.

Remediation for rule failure

Be sure that all containers defined in ContainerDefinitions either omit the Privileged property, or that they set Privileged to false.

The examples that follow show how to implement this remediation.

Amazon ECS Task Definition - Example

Amazon ECS task definition configured with privileged mode deactivated for container definitions. The example is shown in JSON and in YAML.

JSON example

```json
{
  "ECSTaskDefinition": {
    "Type": "AWS::ECS::TaskDefinition",
    "Properties": {
      "ContainerDefinitions": [
        {
          "Essential": true,
          "Image": "alpine:latest",
          "Name": "SampleContainerA"
        },
        {
          "Image": "nginx:latest",
          "Name": "SampleContainerB",
          "Privileged": false
        }
      ],
      "Memory": "512"
    }
  }
}
```

YAML example

```
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    ContainerDefinitions:
    - Essential: true
      Image: alpine:latest
      Name: SampleContainerA
    - Image: nginx:latest
      Name: SampleContainerB
      Privileged: false
    Memory: '512'
```

CT.ECS.PR.11 rule specification

```
# ############################################################################
##       Rule Specification       
# ############################################################################
```
# Rule Identifier:
#     ecs_containers_nonprivileged_check
#
# Description:
#     This control checks whether container definitions in Amazon Elastic Container Service (ECS) task definitions are configured with elevated privileges.
#
# Reports on:
#     AWS::ECS::TaskDefinition
#
# Evaluates:
#     AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#     None
#
# Scenarios:
#     Scenario: 1
#       Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#           And: The input document does not contain an ECS task definition resource
#           Then: SKIP
#     Scenario: 2
#       Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#           And: The input document contains an ECS task definition resource
#           And: 'ContainerDefinitions' property is not present or is an empty list
#           Then: SKIP
#     Scenario: 3
#       Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#           And: The input document contains an ECS task definition resource
#           And: 'ContainerDefinitions' property is present
#           And: One or more containers defined in 'ContainerDefinitions' have 'Privileged' set to bool(true)
#           Then: FAIL
#     Scenario: 4
#       Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#           And: The input document contains an ECS task definition resource
#           And: 'ContainerDefinitions' property is present
#           And: All containers defined in 'ContainerDefinitions' either do not have the 'Privileged' property present or
#           'Privileged' is present and set to bool(false)
#           Then: PASS
#
# Constants
#
let ECS_TASK_DEFINITION_TYPE = "AWS::ECS::TaskDefinition"
let INPUT_DOCUMENT = this
#
# Assignments
#
let ecs_task_definitions = Resources.*[ Type == %ECS_TASK_DEFINITION_TYPE ]
#
# Primary Rules
#
rule ecs_containers_nonprivileged_check when is_cfn_template(%INPUT_DOCUMENT)
  %ecs_task_definitions not empty { 
    check(%ecs_task_definitions.Properties)
    "<<
      [CT.ECS.PR.11]: Require an Amazon ECS container to run as non-privileged
[FIX]: Be sure that all containers defined in 'ContainerDefinitions' either omit the 'Privileged' property, or that they set 'Privileged' to 'false'.

```less
rule ecs_containers_nonprivileged_check when is_cfn_hook(%INPUT_DOCUMENT, %ECS_TASK_DEFINITION_TYPE) {
    check(%INPUT_DOCUMENT.%ECS_TASK_DEFINITION_TYPE.resourceProperties)
    [CT.ECS.PR.11]: Require an Amazon ECS container to run as non-privileged
    [FIX]: Be sure that all containers defined in 'ContainerDefinitions' either omit the 'Privileged' property, or that they set 'Privileged' to 'false'.
}
```

## Parameterized Rules

```less
# Parameterized Rules
#
rule check(ecs_task_definition) {
    %ecs_task_definition [
        filter_container_definitions_is_present(this)
    ]{
        ContainerDefinitions[*] {
            # Scenario 3
            Privileged not exists or
            # Scenario 4
            Privileged == false
        }
    }
}
```

```less
rule filter_container_definitions_is_present(ecs_task_definition) {
    %ecs_task_definition {
        # Scenario 2
        ContainerDefinitions exists
        ContainerDefinitions is_list
        ContainerDefinitions not empty
    }
}
```

## Utility Rules

```less
# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
```

```less
rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

---

### CT.ECS.PR.11 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
Properties:
  ContainerDefinitions:
  - Essential: true
    Image: alpine:latest
    Name: ExampleContainerA
  - Essential: true
    Image: nginx:latest
    Name: ExampleContainerB
  Privileged: false
  Memory: '512'

[CT.ECS.PR.12] Require that Amazon ECS task definitions do not pass secrets as container environment variables

This control checks whether Amazon Elastic Container Service (ECS) task definition container definitions include environment variables named AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY, or ECS_ENGINE_AUTH_DATA.

- **Control objective:** Use strong authentication
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ECS::TaskDefinition
- **AWS CloudFormation guard rule:** [CT.ECS.PR.12 rule specification (p. 703)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ECS.PR.12 rule specification (p. 703)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ECS.PR.12 example templates (p. 705)]

Explanation

701
AWS Systems Manager Parameter Store can help you improve the security posture of your organization. We recommend the Parameter Store as a way to store secrets and credentials, instead of passing them into your container instances or entering them into your source code.

**Usage considerations**

- This control applies only to Amazon ECS task definitions that are configured with container definitions.
- This control evaluates plaintext environment variables configured directly on container definitions.

**Remediation for rule failure**

Omit environment variables with Name set to `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY` or `ECS_ENGINE_AUTH_DATA` from container definitions.

The examples that follow show how to implement this remediation.

**Amazon ECS Task Definition - Example**

Amazon ECS task definition configured to inject sensitive data into a container as an environment variable. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ECSTaskDefinition": {
        "Type": "AWS::ECS::TaskDefinition",
        "Properties": {
            "Memory": "512",
            "ExecutionRoleArn": {
                "Fn::GetAtt": [
                    "ECSTaskExecutionRole",
                    "Arn"
                ]
            },
            "ContainerDefinitions": [
                {
                    "Essential": true,
                    "Image": "nginx:latest",
                    "Name": "SampleContainer",
                    "Environment": [
                        {
                            "Name": "SAMPLE_ENV_VAR",
                            "Value": "sampleValue"
                        }
                    ],
                    "Secrets": [
                        {
                            "Name": "SAMPLE_SENSITIVE_ENV_VAR",
                            "ValueFrom": "arn:aws:ssm:us-east-1:123456789012:parameter/sample_parameter"
                        }
                    ]
                }
            ]
        }
    }
}
```
YAML example

ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Memory: '512'
    ExecutionRoleArn: !GetAtt 'ECSTaskExecutionRole.Arn'
  ContainerDefinitions:
    - Essential: true
      Image: nginx:latest
      Name: SampleContainer
      Environment:
        - Name: SAMPLE_ENV_VAR
          Value: sampleValue
      Secrets:
        - Name: SAMPLESENSITIVE_ENV_VAR
          ValueFrom: arn:aws:ssm:us-east-1:123456789012:parameter/sampleParameter

CT.ECS.PR.12 rule specification

# ###################################################################
##       Rule Specification       ##
# ###################################################################
#
# Rule Identifier:
#   ecs_no_environment_secrets_check
#
# Description:
#   This control checks whether Amazon Elastic Container Service (ECS) task definition
#   container definitions include environment variables named 'AWS_ACCESS_KEY_ID',
#   'AWS_SECRET_ACCESS_KEY', or 'ECS_ENGINE_AUTH_DATA'.
#
# Reports on:
#   AWS::ECS::TaskDefinition
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document does not contain an ECS task definition resource
#   Then: SKIP
# Scenario: 2
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains an ECS task definition resource
#   And: 'ContainerDefinitions' property is not present or is empty
#   Then: SKIP
# Scenario: 3
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains an ECS task definition resource
#   And: 'ContainerDefinitions' property is present
#   And: Containers defined in 'ContainerDefinitions' do not have 'Environment' property
#   Then: SKIP
# Scenario: 4
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document

703
And: The input document contains an ECS task definition resource
And: 'ContainerDefinitions' property is present
And: One or more containers defined in 'ContainerDefinitions' have 'Environment' present
And: 'Environment' property has an entry with 'Name' set to 'AWS_ACCESS_KEY_ID',
'AWS_SECRET_ACCESS_KEY', or
'ECS_ENGINE_AUTH_DATA'
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ECS task definition resource
And: 'ContainerDefinitions' property is present
And: One or more containers defined in 'ContainerDefinitions' have 'Environment' present
And: 'Environment' property does not have an entry with 'Name' set to
'AWS_ACCESS_KEY_ID', 'AWS_SECRET_ACCESS_KEY',
or 'ECS_ENGINE_AUTH_DATA'
Then: PASS

# Constants

let ECS_TASK_DEFINITION_TYPE = "AWS::ECS::TaskDefinition"
let INPUT_DOCUMENT = this
let RESTRICTED_ENVIRONMENT_VARIABLES = ["AWS_ACCESS_KEY_ID", "AWS_SECRET_ACCESS_KEY",
"ECS_ENGINE_AUTH_DATA"]

# Assignments

let ecs_task_definitions = Resources.*[ Type == %ECS_TASK_DEFINITION_TYPE ]

# Primary Rules

rule ecs_no_environment_secrets_check when is_cfn_template(%INPUT_DOCUMENT)
%ecs_task_definitions not empty {
  check(%ecs_task_definitions.Properties)
  <<
  [CT.ECS.PR.12]: Require that Amazon ECS task definitions do not pass secrets as container environment variables
  [FIX]: Omit environment variables with 'Name' set to 'AWS_ACCESS_KEY_ID',
  'AWS_SECRET_ACCESS_KEY' or 'ECS_ENGINE_AUTH_DATA' from container definitions.
  >>
}

rule ecs_no_environment_secrets_check when is_cfn_hook(%INPUT_DOCUMENT,
%ECS_TASK_DEFINITION_TYPE) {
  check(%INPUT_DOCUMENT.%ECS_TASK_DEFINITION_TYPE.resourceProperties)
  <<
  [CT.ECS.PR.12]: Require that Amazon ECS task definitions do not pass secrets as container environment variables
  [FIX]: Omit environment variables with 'Name' set to 'AWS_ACCESS_KEY_ID',
  'AWS_SECRET_ACCESS_KEY' or 'ECS_ENGINE_AUTH_DATA' from container definitions.
  >>
}

# Parameterized Rules

rule check(ecs_task_definition) {
  %ecs_task_definition [filter_container_definitions_is_present(this)]
  ContainerDefinitions[filter_environment_is_present(this)
CT.ECS.PR.12 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service:
            - ecs-tasks.amazonaws.com
          Action:
            - sts:AssumeRole
          Path: /
ManagedPolicyArns:
- arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy

Policies:
- PolicyName: ECSTaskPolicy
  PolicyDocument:
  Version: '2012-10-17'
  Statement:
  - Effect: Allow
    Action:
    - ssm:GetParameters
  Resource: arn:aws:ssm:us-east-1:123456789012:parameter/example_parameter

ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
  Memory: '512'
  ExecutionRoleArn:
  - Fn::GetAtt: [ ECSTaskExecutionRole, Arn ]
  ContainerDefinitions:
  - Essential: true
    Image: nginx:latest
    Name: ExampleContainer
    Environment:
    - Name: EXAMPLE_ENV_VAR
      Value: exampleValue
    Secrets:
    - Name: EXAMPLE_SENSITIVE_ENV_VAR
      ValueFrom: arn:aws:ssm:us-east-1:123456789012:parameter/example_parameter

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
  Memory: '512'
  ContainerDefinitions:
  - Essential: true
    Image: nginx:latest
    Name: ExampleContainerA
    Environment:
    - Name: AWS_ACCESS_KEY_ID
      Value: exampleKey
    - Name: AWS_SECRET_ACCESS_KEY
      Value: exampleSecretKey
    - Image: alpine:latest
      Name: ExampleContainerB

Amazon Elastic File System controls

Topics
- [CT.ELASTICFILESYSTEM.PR.1] Require an Amazon EFS file system to encrypt file data at rest using AWS KMS (p. 707)
- [CT.ELASTICFILESYSTEM.PR.2] Require an Amazon EFS volume to have an automated backup plan (p. 710)
- [CT.ELASTICFILESYSTEM.PR.3] Require Amazon EFS access points to have a root directory (p. 714)
• [CT.ELASTICFILESYSTEM.PR.4] Require Amazon EFS access points to enforce a user identity (p. 718)

[CT.ELASTICFILESYSTEM.PR.1] Require an Amazon EFS file system to encrypt file data at rest using AWS KMS

This control checks whether an Amazon Elastic File System (Amazon EFS) file system is configured to encrypt file data using AWS KMS.

• Control objective: Encrypt data at rest
• Implementation: AWS CloudFormation Guard Rule
• Control behavior: Proactive
• Resource types: AWS::EFS::FileSystem
• AWS CloudFormation guard rule: CT.ELASTICFILESYSTEM.PR.1 rule specification (p. 708)

Details and examples

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ELASTICFILESYSTEM.PR.1 rule specification (p. 708)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.ELASTICFILESYSTEM.PR.1 example templates (p. 710)

Explanation

For an added layer of security for your sensitive data in Amazon EFS, you should create encrypted file systems. Amazon EFS supports encryption for file systems at rest. You can enable encryption of data at rest when you create an Amazon EFS file system.

Usage considerations

• This control requires only the Encrypted property to be set to true, and it does not require the KmsKeyId property to be provided.
• If the KmsKeyId property is not provided, the default AWS KMS key for Amazon EFS, /aws/elasticfilesystem, is used to protect the encrypted file system.

Remediation for rule failure

Set Encrypted to true and optionally set KmsKeyId to a valid AWS KMS key identifier.

The examples that follow show how to implement this remediation.

Amazon EFS File System - Example One

Amazon EFS file system configured with encryption enabled, by means of the default AWS KMS key for Amazon EFS. The example is shown in JSON and in YAML.

JSON example

```json
{
    "EFSFileSystem": {
        "Type": "AWS::EFS::FileSystem",
        "Properties": {
            "Encrypted": true
        }
    }
}
```
YAML example

EFSFileSystem:
  Type: AWS::EFS::FileSystem
  Properties:
    Encrypted: true

The examples that follow show how to implement this remediation.

Amazon EFS File System - Example Two

Amazon EFS file system configured with encryption enabled, by means of a customer-managed AWS KMS key. The example is shown in JSON and in YAML.

JSON example

```
{
  "EFSFileSystem": {
    "Type": "AWS::EFS::FileSystem",
    "Properties": {
      "Encrypted": true,
      "KmsKeyId": {
        "Ref": "KMSKey"
      }
    }
  }
}
```

YAML example

```
EFSFileSystem:
  Type: AWS::EFS::FileSystem
  Properties:
    Encrypted: true
    KmsKeyId: !Ref 'KMSKey'
```

CT.ELASTICFILESYSTEM.PR.1 rule specification

```
# ################################################################################
##       Rule Specification        ##
# ################################################################################
# Rule Identifier:
#   efs_encrypted_check
#
# Description:
#  This control checks whether an Amazon Elastic File System (Amazon EFS) file system is configured to encrypt file data using AWS KMS.
#
# Reports on:
```
AWS Control Tower User Guide
Proactive controls

# AWS::EFS::FileSystem
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#            And: The input document does not contain any Amazon EFS file system resources
#            Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#            And: The input document contains an Amazon EFS file system resource
#            And: 'Encrypted' is not present
#            Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#            And: The input document contains an Amazon EFS file system resource
#            And: 'Encrypted' is present and set to bool(false)
#            Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#            And: The input document contains an Amazon EFS file system resource
#            And: 'Encrypted' is present and set to bool(true)
#            Then: PASS
#
# Constants
#
let RESOURCE_TYPE = "AWS::EFS::FileSystem"
let INPUT_DOCUMENT = this
#
# Assignments
#
let efs_file_systems = Resources.*[ Type == %RESOURCE_TYPE ]
#
# Primary Rules
#
rule efs_encrypted_check when is_cfn_template(%INPUT_DOCUMENT)
  %efs_file_systems not empty {
    check(%efs_file_systems.Properties)
    <<
    [CT.ELASTICFILESYSTEM.PR.1]: Require an Amazon EFS file system to encrypt file data at rest using AWS KMS
    [FIX]: Set 'Encrypted' to 'true' and optionally set 'KmsKeyId' to a valid AWS KMS key identifier.
    >>
}
rule efs_encrypted_check when is_cfn_hook(%INPUT_DOCUMENT, %RESOURCE_TYPE) {
  check(%INPUT_DOCUMENT.%RESOURCE_TYPE.resourceProperties)
  <<
  [CT.ELASTICFILESYSTEM.PR.1]: Require an Amazon EFS file system to encrypt file data at rest using AWS KMS
  [FIX]: Set 'Encrypted' to 'true' and optionally set 'KmsKeyId' to a valid AWS KMS key identifier.
  >>
}
### Parameterized Rules

```ruby
# Scenario 2
Encrypted exists
# Scenario 3 and 4
Encrypted == true
```

#### Utility Rules

```ruby
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

---

**CT.ELASTICFILESYSTEM.PR.1 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```ruby
Resources:
  EFSFileSystem:
    Type: AWS::EFS::FileSystem
    Properties:
      Encrypted: true
```

**FAIL Example** - Use this template to verify that the control prevents non-compliant resource creation.

```ruby
Resources:
  EFSFileSystem:
    Type: AWS::EFS::FileSystem
    Properties:
      Encrypted: false
```

**[CT.ELASTICFILESYSTEM.PR.2] Require an Amazon EFS volume to have an automated backup plan**

This control checks whether your Amazon Elastic File System (Amazon EFS) file system has been configured with automatic backups through AWS Backup.
• **Control objective:** Improve resiliency
• **Implementation:** AWS CloudFormation Guard Rule
• **Control behavior:** Proactive
• **Resource types:** AWS::EFS::FileSystem
• **AWS CloudFormation guard rule:** [CT.ELASTICFILESYSTEM.PR.2 rule specification (p. 711)]

**Details and examples**

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the:  
  [CT.ELASTICFILESYSTEM.PR.2 rule specification (p. 711)]
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: 
  [CT.ELASTICFILESYSTEM.PR.2 example templates (p. 713)]

**Explanation**

Including Amazon EFS file systems in the backup plans helps you to protect your data from deletion and data loss.

**Remediation for rule failure**

Enable automatic backups by setting `BackupPolicy.Status` to `ENABLED`.

The examples that follow show how to implement this remediation.

**Amazon EFS File System - Example**

Amazon EFS file system configured with automatic backups enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "EFSFileSystem": {
      "Type": "AWS::EFS::FileSystem",
      "Properties": {
         "BackupPolicy": {
            "Status": "ENABLED"
         }
      }
   }
}
```

**YAML example**

```yaml
EFSFileSystem:
  Type: AWS::EFS::FileSystem
  Properties:
    BackupPolicy:
      Status: ENABLED
```

**CT.ELASTICFILESYSTEM.PR.2 rule specification**

711
# Proactive controls

## Rule Specification

# Rule Identifier:
# efs_automatic_backups_enabled_check

# Description:
# This control checks whether your Amazon Elastic File System (Amazon EFS) file system has been configured with automatic backups through AWS Backup.

# Reports on:
# AWS::EFS::FileSystem

# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document does not contain any Amazon EFS file system resources
#   Then: SKIP
# Scenario: 2
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains an Amazon EFS file system resource
#   And: 'BackupPolicy' is not present
#   Then: FAIL
# Scenario: 3
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains an Amazon EFS file system resource
#   And: 'BackupPolicy' is present and 'Status' is set to 'DISABLED'
#   Then: FAIL
# Scenario: 4
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains an Amazon EFS file system resource
#   And: 'BackupPolicy' is present and 'Status' is set to 'ENABLED'
#   Then: PASS

# Constants
#
let RESOURCE_TYPE = "AWS::EFS::FileSystem"
let INPUT_DOCUMENT = this

# Assignments
#
let efs_file_systems = Resources.*[ Type == %RESOURCE_TYPE ]

# Primary Rules
#
rule efs_automatic_backups_enabled_check when is_cfn_template(this)
  %efs_file_systems not empty {
    check(%efs_file_systems.Properties)
    <=
    [CT.ELASTICFILESYSTEPR.2]: Require an Amazon EFS volume to have an automated backup plan
    [FIX]: Enable automatic backups by setting 'BackupPolicy.Status' to 'ENABLED'.
AWS Control Tower User Guide
Proactive controls

CT.ELASTICFILESYSTEM.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
EFSFileSystem:
    Type: AWS::EFS::FileSystem
    Properties:
        BackupPolicy:
            Status: ENABLED

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
EFSFileSystem:
    Type: AWS::EFS::FileSystem
    Properties:
        BackupPolicy:
            Status: ENABLED
[CT.ELASTICFILESYSTEM.PR.3] Require Amazon EFS access points to have a root directory

This control checks whether your Amazon Elastic File System (Amazon EFS) access points are configured to enforce a root directory.

- **Control objective:** Enforce least privilege
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EFS::AccessPoint
- **AWS CloudFormation guard rule:** [CT.ELASTICFILESYSTEM.PR.3 rule specification](#) (p. 715)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICFILESYSTEM.PR.3 rule specification](#) (p. 715)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICFILESYSTEM.PR.3 example templates](#) (p. 717)

**Explanation**

When you enforce a root directory, the NFS client at the access point uses the root directory configured on the access point, instead of the file system's root directory. Enforcing a root directory for an access point helps restrict data access by ensuring that users of the access point can reach only the files of the specified subdirectory.

**Remediation for rule failure**

Provide a `RootDirectory.Path` configuration with a value for `Path` that does not equal `/`.

The examples that follow show how to implement this remediation.

**Amazon EFS Access Point - Example**

Amazon EFS access point configured with a root directory set to a specific subdirectory. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "EFSAccessPoint": {
        "Type": "AWS::EFS::AccessPoint",
        "Properties": {
```
YAML example

EFSAccessPoint:
  Type: AWS::EFS::AccessPoint
  Properties:
    FileSystemId: !Ref 'EFSFileSystem'
    RootDirectory:
      Path: /dir1/child1

CT.ELASTICFILESYSTEM.PR.3 rule specification

```yaml
# ##############################################################################
##       Rule Specification        ##
# ##############################################################################
#
# Rule Identifier:
#   efs_access_point_enforce_root_directory_check
#
# Description:
#   This control checks whether your Amazon Elastic File System (Amazon EFS) access points
#   are configured to enforce a root directory.
#
# Reports on:
#   AWS::EFS::AccessPoint
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document does not contain any Amazon EFS access point resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document contains an Amazon EFS access point resource
#     And: 'RootDirectory' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document contains an Amazon EFS access point resource
#     And: 'RootDirectory' has been provided
```
And: 'Path' within 'RootDirectory' has not been provided or has been provided with an empty string value
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon EFS access point resource
And: 'RootDirectory' has been provided
And: 'Path' within 'RootDirectory' been provided with a value of '/'
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon EFS access point resource
And: 'RootDirectory' has been provided
And: 'Path' within 'RootDirectory' been provided with a non-empty string value not equal to '/'
Then: PASS

Constants
let EFS_ACCESS_POINT_TYPE = "AWS::EFS::AccessPoint"
let INPUT_DOCUMENT = this

Assignments
let efs_access_points = Resources.*[ Type == %EFS_ACCESS_POINT_TYPE ]

Primary Rules
rule efs_access_point_enforce_root_directory_check when is_cfn_template(%INPUT_DOCUMENT)
%efs_access_points not empty {
  check(%efs_access_points.Properties)
  # Scenario 2
  RootDirectory exists
  RootDirectory {
    # Scenario 3, 4 and 5
    Path exists
    check_is_string_and_not_empty(Path)
    <![CT.ELASTICFILESYSTEM.PR.3]: Require Amazon EFS access points to have a root directory
    [FIX]: Provide a 'RootDirectory.Path' configuration with a value for 'Path' that does not equal '/'.
    ]>
  }%efs_access_points not empty {
  check(%efs_access_points.Properties)
  # Scenario 2
  RootDirectory exists
  RootDirectory {
    # Scenario 3, 4 and 5
    Path exists
    check_is_string_and_not_empty(Path)
    <![CT.ELASTICFILESYSTEM.PR.3]: Require Amazon EFS access points to have a root directory
    [FIX]: Provide a 'RootDirectory.Path' configuration with a value for 'Path' that does not equal '/'.
    ]>
  }
}

Parameterized Rules
rule check(efs_access_points) {
  %efs_access_points {
    # Scenario 2
    RootDirectory exists
    RootDirectory {
      # Scenario 3, 4 and 5
      Path exists
      check_is_string_and_not_empty(Path)
      <![CT.ELASTICFILESYSTEM.PR.3]: Require Amazon EFS access points to have a root directory
      [FIX]: Provide a 'RootDirectory.Path' configuration with a value for 'Path' that does not equal '/'.
      ]>
    }
  }
}
CT.ELASTICFILESYSTEM.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
EFSAccessPoint:
  Type: AWS::EFS::AccessPoint
  Properties:
    FileSystemId:
      Ref: EFSFileSystem
    RootDirectory:
      Path: /dir1/child1
EFSFileSystem:
  Type: AWS::EFS::FileSystem
  Properties: {}
[CT.ELASTICFILESYSTEM.PR.4] Require Amazon EFS access points to enforce a user identity

This control checks whether your Amazon Elastic File System (Amazon EFS) access points are configured to enforce a user identity.

- **Control objective:** Enforce least privilege
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EFS::AccessPoint
- **AWS CloudFormation guard rule:** [CT.ELASTICFILESYSTEM.PR.4 rule specification (p. 719)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICFILESYSTEM.PR.4 rule specification (p. 719)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICFILESYSTEM.PR.4 example templates (p. 721)]

**Explanation**

Amazon EFS access points are application-specific entry points into an Amazon EFS file system that make it easier to manage application access to shared datasets. Access points can enforce a user identity, including the user’s POSIX groups, for all file system requests that are made through the access point. Access points also can enforce a different root directory for the file system, so that clients gain access only to data in the specified directory or its subdirectories.

**Remediation for rule failure**

Provide a PosixUser configuration with a POSIX user ID (Uid) and POSIX group ID (Gid).

The examples that follow show how to implement this remediation.

**Amazon EFS Access Point - Example**

Amazon EFS access point configured to enforce a user identity for all file system requests made through the access point. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "EFSAccessPoint": {
    "Type": "AWS::EFS::AccessPoint",
    "Properties": {
      "FileSystemId": {
        "Ref": "EFSFileSystem"
      },
      "PosixUser": {
        "Uid": "111",
        "Gid": "222"
      }
    }
  }
}
```
YAML example

EFSAccessPoint:
  Type: AWS::EFS::AccessPoint
  Properties:
    FileSystemId: !Ref 'EFSFileSystem'
    PosixUser:
      Uid: '111'
      Gid: '222'

CT.ELASTICFILESYSTEM.PR.4 rule specification

# ****************************
## Rule Specification        ##
# ****************************

# Rule Identifier:
#   efs_access_point_enforce_user_identity_check
#
# Description:
#   This control checks whether your Amazon Elastic File System (Amazon EFS) access points
#   are configured to enforce a user identity.
#
# Reports on:
#   AWS::EFS::AccessPoint
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document does not contain any Amazon EFS access point resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document contains an Amazon EFS access point resource
#     And: 'PosixUser' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document contains an Amazon EFS access point resource
#     And: 'PosixUser' has been provided
#     And: 'Uid' within 'PosixUser' has not been provided or has been provided with an
#     empty string value
#     And: 'Gid' within 'PosixUser' has not been provided or has been provided with an
#     empty string value
#     Then: FAIL
#   Scenario: 4
#     Given: The input document contains an Amazon EFS access point resource
#     And: 'PosixUser' has been provided
#     And: 'Uid' within 'PosixUser' has not been provided or has been provided with an
#     empty string value
#     Then: FAIL
#
And: The input document contains an Amazon EFS access point resource
And: 'PosixUser' has been provided
And: 'Uid' within 'PosixUser' has been provided with a non-empty string value
And: 'Gid' within 'PosixUser' has been provided with a non-empty string value
Then: PASS

# Constants
#
let EFS_ACCESS_POINT_TYPE = "AWS::EFS::AccessPoint"
let INPUT_DOCUMENT = this
#
# Assignments
#
let efs_access_points = Resources.*[ Type == %EFS_ACCESS_POINT_TYPE ]
#
# Primary Rules
#
rule efs_access_point_enforce_user_identity_check when is_cfn_template(this)
    %efs_access_points not empty {
        check(%efs_access_points.Properties)
        <<
        [CT.ELASTICFILESYSTEM.PR.4]: Require Amazon EFS access points to enforce a user
        identity
        [FIX]: Provide a 'PosixUser' configuration with a POSIX user ID ('Uid') and POSIX
        group ID ('Gid').
        >>
    }

rule efs_access_point_enforce_user_identity_check when is_cfn_hook(%INPUT_DOCUMENT,
    %EFS_ACCESS_POINT_TYPE) {
    check(%INPUT_DOCUMENT.%EFS_ACCESS_POINT_TYPE.resourceProperties)
    <<
    [CT.ELASTICFILESYSTEM.PR.4]: Require Amazon EFS access points to enforce a user
    identity
    [FIX]: Provide a 'PosixUser' configuration with a POSIX user ID ('Uid') and POSIX
    group ID ('Gid').
    >>
}
#
# Parameterized Rules
#
rule check(efs_access_points) {
    %efs_access_points {
        # Scenario 2
        PosixUser exists
        PosixUser {
            # Scenario 3 and 4
            Uid exists
            check_is_string_and_not_empty(Uid)
            Gid exists
            check_is_string_and_not_empty(Gid)
        }
    }
}
#
# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
CT.ELASTICFILESYSTEM.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  EFSAccessPoint:
    Type: AWS::EFS::AccessPoint
    Properties:
      FileSystemId:
        Ref: EFSFileSystem
      PosixUser:
        Uid: '111'
        Gid: '222'
  EFSFileSystem:
    Type: AWS::EFS::FileSystem
    Properties: {}
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
  EFSAccessPoint:
    Type: AWS::EFS::AccessPoint
    Properties:
      FileSystemId:
        Ref: EFSFileSystem
  EFSFileSystem:
    Type: AWS::EFS::FileSystem
    Properties: {}
```

Amazon Elastic Kubernetes Service (EKS) controls

Topics
- [CT.EKS.PR.1] Require an Amazon EKS cluster to be configured with public access disabled to the cluster Kubernetes API server endpoint (p. 722)
- [CT.EKS.PR.2] Require an Amazon EKS cluster to be configured with secret encryption using AWS Key Management Service (KMS) keys (p. 727)
[CT.EKS.PR.1] Require an Amazon EKS cluster to be configured with public access disabled to the cluster Kubernetes API server endpoint

This control checks whether an Amazon Elastic Kubernetes Service (EKS) cluster endpoint disallows public access to the cluster Kubernetes API server endpoint.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EKS::Cluster
- **AWS CloudFormation guard rule:** [CT.EKS.PR.1 rule specification](p. 723)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.EKS.PR.1 rule specification](p. 723)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.EKS.PR.1 example templates](p. 725)

**Explanation**

When you create a new cluster, Amazon Elastic Kubernetes Service (EKS) creates an endpoint for the managed Kubernetes API server, which you can use to communicate with your cluster using Kubernetes management tools, such as `kubectl`. By default, this API server endpoint is public to the internet, and access to the API server is secured using a combination of AWS Identity and Access Management (IAM) along with native Kubernetes Role-Based Access Control (RBAC). Enabling private access to the Kubernetes API server ensures that all communication between your nodes and the API server stays within your VPC. You can limit the IP addresses that have access to your API server from the internet, or you can completely disallow internet access to the API server.

**Remediation for rule failure**

Set the value of the EndpointPublicAccess parameter to false and the value of the EndpointPrivateAccess parameter to true.

The examples that follow show how to implement this remediation.

**Amazon EKS Cluster - Example**

An Amazon EKS cluster configured with public access disabled to the cluster's Kubernetes API server endpoint. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "EKSCluster": {
      "Type": "AWS::EKS::Cluster",
      "Properties": {
         "RoleArn": {
            "Fn::GetAtt": "EKSClusterRole.Arn"
         },
         "ResourcesVpcConfig": {
            "SubnetIds": [
               "Ref": "SubnetOne"
            ]
      }
   }
}
```
YAML example

EKSCluster:
  Type: AWS::EKS::Cluster
  Properties:
    RoleArn: !GetAtt 'EKSClusterRole.Arn'
    ResourcesVpcConfig:
      SubnetIds:
        - !Ref 'SubnetOne'
        - !Ref 'SubnetTwo'
      EndpointPublicAccess: false
      EndpointPrivateAccess: true

CT.EKS.PR.1 rule specification

# ####################################################################
##       Rule Specification        
# ####################################################################
#
# Rule Identifier:
#   eks_endpoint_no_public_access_check
#
# Description:
#   This control checks whether an Amazon Elastic Kubernetes Service (EKS) cluster endpoint disallows public access to the cluster Kubernetes API server endpoint.
#
# Reports on:
#   AWS::EKS::Cluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any Amazon EKS cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Amazon EKS cluster resource
#     And: 'EndpointPublicAccess' in 'ResourcesVpcConfig' has not been provided
#     And: 'EndpointPrivateAccess' in 'ResourcesVpcConfig' has not been provided
#     Then: FAIL
# Scenario: 3
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#       And: The input document contains an Amazon EKS cluster resource
#       And: 'EndpointPrivateAccess' in 'ResourcesVpcConfig' has not been provided
#       And: 'EndpointPublicAccess' in 'ResourcesVpcConfig' has not been provided or has
#            been provided and set to a
#            value other than bool(false)
#       Then: FAIL
# Scenario: 4
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#       And: The input document contains an Amazon EKS cluster resource
#       And: 'EndpointPublicAccess' in 'ResourcesVpcConfig' has not been provided or has
#            been provided and set to a
#            value other than bool(false)
#       And: 'EndpointPrivateAccess' in 'ResourcesVpcConfig' has been provided and set a
#            value other than bool(true)
#       Then: FAIL
# Scenario: 5
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#       And: The input document contains an Amazon EKS cluster resource
#       And: 'EndpointPublicAccess' in 'ResourcesVpcConfig' has not been provided or has
#            been provided and set to bool(false)
#       And: 'EndpointPrivateAccess' in 'ResourcesVpcConfig' has been provided and set to
#            bool(true)
#       Then: PASS

# Constants
#
let EKS_CLUSTER_TYPE = "AWS::EKS::Cluster"
let INPUT_DOCUMENT = this

# Assignments
#
let eks_clusters = Resources.*[ Type == %EKS_CLUSTER_TYPE ]

# Primary Rules
#
rule eks_endpoint_no_public_access_check when is_cfn_template(%INPUT_DOCUMENT)
%eks_clusters not empty {
    check(%eks_clusters.Properties)
    <<<
    [CT.EKS.PR.1]: Require an Amazon EKS cluster to be configured with public access
disabled to the cluster Kubernetes API server endpoint.
    [FIX]: Set the value of the 'EndpointPublicAccess' parameter to false and the value
of the 'EndpointPrivateAccess' parameter to true.
    >>>
}
rule eks_endpoint_no_public_access_check when is_cfn_hook(%INPUT_DOCUMENT,
%EKS_CLUSTER_TYPE) {
    check(%INPUT_DOCUMENT.%EKS_CLUSTER_TYPE.resourceProperties)
    <<<
    [CT.EKS.PR.1]: Require an Amazon EKS cluster to be configured with public access
disabled to the cluster Kubernetes API server endpoint.
    [FIX]: Set the value of the 'EndpointPublicAccess' parameter to false and the value
of the 'EndpointPrivateAccess' parameter to true.
    >>>
}

#
# Parameterized Rules

```yaml
# Parameterized Rules

# Rule check(eks_cluster) {
  %eks_cluster {
    ResourcesVpcConfig exists
    ResourcesVpcConfig is_struct
    ResourcesVpcConfig {
      # Scenarios 2, 3, 4 and 5
      EndpointPublicAccess exists
      EndpointPublicAccess == false
      EndpointPrivateAccess exists
      EndpointPrivateAccess == true
    }
  }
}

# Utility Rules

# Rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

CT.EKS.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:

<table>
<thead>
<tr>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VPC:</strong></td>
</tr>
<tr>
<td>Type: AWS::EC2::VPC</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>CidrBlock: 10.0.0.0/16</td>
</tr>
<tr>
<td>EnableDnsSupport: 'true'</td>
</tr>
<tr>
<td>EnableDnsHostnames: 'true'</td>
</tr>
<tr>
<td>SubnetOne:</td>
</tr>
<tr>
<td>Type: AWS::EC2::Subnet</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>VpcId:</td>
</tr>
<tr>
<td>Ref: VPC</td>
</tr>
<tr>
<td>CidrBlock: 10.0.0.0/24</td>
</tr>
<tr>
<td>AvailabilityZone:</td>
</tr>
<tr>
<td>Fn::Select:</td>
</tr>
<tr>
<td>- 0</td>
</tr>
<tr>
<td>- Fn::GetAZs: ''</td>
</tr>
<tr>
<td>SubnetTwo:</td>
</tr>
<tr>
<td>Type: AWS::EC2::Subnet</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>VpcId:</td>
</tr>
<tr>
<td>Ref: VPC</td>
</tr>
<tr>
<td>CidrBlock: 10.0.1.0/24</td>
</tr>
<tr>
<td>AvailabilityZone:</td>
</tr>
</tbody>
</table>
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: ''
SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
        - 1
        - Fn::GetAZs: ''
EKSClusterRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: eks.amazonaws.com
            Action: sts:AssumeRole
          Path: /
          ManagedPolicyArns:
            - arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
EKSCluster:
  Type: AWS::EKS::Cluster
  Properties:
    RoleArn:
      Fn::GetAtt: EKSClusterRole.Arn
    ResourcesVpcConfig:
      SubnetIds:
        - Ref: SubnetOne
        - Ref: SubnetTwo
      EndpointPublicAccess: false
      EndpointPrivateAccess: true
[CT.EKS.PR.2] Require an Amazon EKS cluster to be configured with secret encryption using AWS Key Management Service (KMS) keys

This control checks whether Amazon Elastic Kubernetes Service (Amazon EKS) clusters are configured to use Kubernetes secrets encrypted with AWS Key Management Service (KMS) keys.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EKS::Cluster
- **AWS CloudFormation guard rule:** [CT.EKS.PR.2 rule specification (p. 729)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.EKS.PR.2 rule specification (p. 729)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.EKS.PR.2 example templates (p. 732)]

**Explanation**

Kubernetes secrets store sensitive information, such as user certificates, passwords, or API keys. Encrypting Kubernetes secrets at rest bolsters the security of your EKS clusters.

**Usage considerations**

- For a cluster that uses KMS Envelope Encryption, kms:CreateGrant permissions are required. The condition kms:GrantIsForAWSResource is not supported for the CreateCluster action, and this condition should not be given in KMS policies to control kms:CreateGrant permissions for users performing CreateCluster operations.

**Remediation for rule failure**

Provide an EncryptionConfig configuration with a list of Resources that contains secrets and a Provider configuration containing a KeyArn.
The examples that follow show how to implement this remediation.

**Amazon EKS cluster - Example**

Amazon EKS cluster configured to have Kubernetes secrets encrypted using Amazon Elastic Kubernetes Service (KMS) keys. The example is shown in JSON and in YAML.

**JSON example**

```

{
    "EKSCluster": {
        "Type": "AWS::EKS::Cluster",
        "Properties": {
            "RoleArn": {
                "Fn::GetAtt": [
                    "EKSClusterRole",
                    "Arn"
                ]
            },
            "ResourcesVpcConfig": {
                "SubnetIds": [
                    { "Ref": "SubnetOne" },
                    { "Ref": "SubnetTwo" }
                ],
                "EndpointPublicAccess": false,
                "EndpointPrivateAccess": true
            },
            "Logging": {
                "ClusterLogging": {
                    "EnabledTypes": [
                        { "Type": "api" },
                        { "Type": "audit" },
                        { "Type": "authenticator" },
                        { "Type": "controllerManager" },
                        { "Type": "scheduler" }
                    ]
                },
                "EncryptionConfig": [
                    { "Resources": ["secrets"],
                    "Provider": {
                        "KeyArn": {
                            "Fn::GetAtt": [
                                "KMSKey",
                                "Arn"
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```
YAML example

EKSCluster:
  Type: AWS::EKS::Cluster
  Properties:
    RoleArn: !GetAtt 'EKSClusterRole.Arn'
    ResourcesVpcConfig:
      SubnetIds:
        - !Ref 'SubnetOne'
        - !Ref 'SubnetTwo'
    EndpointPublicAccess: false
    EndpointPrivateAccess: true
  Logging:
    ClusterLogging:
      EnabledTypes:
        - Type: api
        - Type: audit
        - Type: authenticator
        - Type: controllerManager
        - Type: scheduler
    EncryptionConfig:
      Resources:
        - secrets
      Provider:
        KeyArn: !GetAtt 'KMSKey.Arn'

CT.EKS.PR.2 rule specification

# ####################################################################
#                  Rule Specification                                 #
# ####################################################################
#
# Rule Identifier:     # eks_secrets_encrypted_check
#
# Description:        # This control checks whether Amazon Elastic Kubernetes Service (Amazon EKS) clusters are configured to use Kubernetes secrets encrypted with AWS Key Management Service (KMS) keys.
#
# Reports on:         # AWS::EKS::Cluster
#                      # AWS CloudFormation, AWS CloudFormation hook
#                      # Rule Parameters: None
#                      # Scenarios: Scenario: 1
#                      # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document does not contain any Amazon EKS cluster resources
Then: SKIP

Scenario: 2
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon EKS cluster resource
And: 'EncryptionConfig' has not been provided or provided as an empty list
Then: FAIL

Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon EKS cluster resource
And: 'EncryptionConfig' has been provided as a non-empty list
And: There are no entries in 'EncryptionConfig' where 'Resources' has been provided as a non-empty list with at least one value equal to 'secrets'
And: For the same entry in 'EncryptionConfig', where 'KeyArn' in 'Provider' has been provided as a non-empty string or valid local reference to a KMS key or key alias
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon EKS cluster resource
And: 'EncryptionConfig' has been provided as a non-empty list
And: For at least one entry in 'EncryptionConfig', 'Resources' has been provided as a non-empty list with at least one value equal to 'secrets'
And: For the same entry in 'EncryptionConfig', 'KeyArn' in 'Provider' has been provided as a non-empty string or valid local reference to a KMS key or key alias
Then: PASS
AWS Control Tower User Guide
Proactive controls
}
#
# Parameterized Rules
#
rule check(eks_cluster) {
%eks_cluster {
# Scenario 2
EncryptionConfig exists
EncryptionConfig is_list
EncryptionConfig not empty
# Scenario 3 and 4
some EncryptionConfig[*] {
Resources exists
Resources is_list
Resources not empty
some Resources[*] == "secrets"

}

}

}

Provider exists
Provider is_struct
Provider {
KeyArn exists
check_is_string_and_not_empty(KeyArn) or
check_local_references(%INPUT_DOCUMENT, KeyArn, "AWS::KMS::Key") or
check_local_references(%INPUT_DOCUMENT, KeyArn, "AWS::KMS::Alias")
}

#
# Utility Rules
#
rule is_cfn_template(doc) {
%doc {
AWSTemplateFormatVersion exists
Resources exists
}
}

or

rule is_cfn_hook(doc, RESOURCE_TYPE) {
%doc.%RESOURCE_TYPE.resourceProperties exists
}
rule check_is_string_and_not_empty(value) {
%value {
this is_string
this != /\A\s*\z/
}
}
rule check_local_references(doc, reference_properties, referenced_resource_type) {
%reference_properties {
'Fn::GetAtt' {
query_for_resource(%doc, this[0], %referenced_resource_type)
<<Local Stack reference was invalid>>
} or Ref {
query_for_resource(%doc, this, %referenced_resource_type)
<<Local Stack reference was invalid>>
}
}
}
rule query_for_resource(doc, resource_key, referenced_resource_type) {
let referenced_resource = %doc.Resources[ keys == %resource_key ]

731


CT.EKS.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
%referenced_resource not empty
%referenced_resource {
  Type == %referenced_resource_type
}
}

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
  SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/24
      SubnetOne:
        Type: AWS::EC2::Subnet
        Properties:
          VpcId:
            Ref: VPC
        CidrBlock: 10.0.1.0/24
        AvailabilityZone:
          Fn::Select:
            - 1
          - Fn::GetAZs: ''
  EKSClusterRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: eks.amazonaws.com
            Action: sts:AssumeRole
          - Path: /
            ManagedPolicyArns:
              - arn:aws:iam::aws:policy/AmazonEKSClusterPolicy
  KMSKey:
    Type: AWS::KMS::Key
    Properties:
      PendingWindowInDays: 7
      KeyPolicy:
        Version: 2012-10-17
        Id: example-key-policy
        Statement:
          - Sid: Enable IAM User Permissions
```

Effect: Allow
Principal:
  AWS:
    Fn::Sub: arn:${AWS::Partition}:iam::${AWS::AccountId}:root
  Action: kms:*
  Resource: '*'
KeySpec: SYMMETRIC_DEFAULT
EKSCluster:
  Type: AWS::EKS::Cluster
  Properties:
    RoleArn:
      Fn::GetAtt:
        - EKSClusterRole
        - Arn
    ResourcesVpcConfig:
      SubnetIds:
        - Ref: SubnetOne
        - Ref: SubnetTwo
    EndpointPublicAccess: false
    EndpointPrivateAccess: true
Logging:
  ClusterLogging:
    EnabledTypes:
      - Type: api
      - Type: audit
      - Type: authenticator
      - Type: controllerManager
      - Type: scheduler
  EncryptionConfig:
    - Resources:
      - secrets
    Provider:
      KeyArn:
        Fn::GetAtt: [KMSKey, Arn]

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
      - 0
      - Fn::GetAZs: ''
SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
Elastic Load Balancing controls

Topics

- [CT.ELASTICLOADBALANCING.PR.1] Require any application load balancer listener default actions to redirect all HTTP requests to HTTPS (p. 735)
- [CT.ELASTICLOADBALANCING.PR.2] Require any Amazon ELB application or network load balancer to have an AWS Certificate Manager certificate (p. 741)
- [CT.ELASTICLOADBALANCING.PR.3] Require any application load balancer to have defensive or strictest desync mitigation mode activated (p. 748)
- [CT.ELASTICLOADBALANCING.PR.4] Require that any application load balancer must be configured to drop HTTP headers (p. 754)
- [CT.ELASTICLOADBALANCING.PR.5] Require that application load balancer deletion protection is activated (p. 760)
- [CT.ELASTICLOADBALANCING.PR.6] Require that application and network load balancer access logging is activated (p. 764)
- [CT.ELASTICLOADBALANCING.PR.7] Require any classic load balancer to have multiple Availability Zones configured (p. 773)
- [CT.ELASTICLOADBALANCING.PR.8] Require any classic load balancer SSL/HTTPS listener to have a certificate provided by AWS Certificate Manager (p. 781)
- [CT.ELASTICLOADBALANCING.PR.9] Require that an AWS ELB Application or Classic Load Balancer listener is configured with HTTPS or TLS termination (p. 787)
- [CT.ELASTICLOADBALANCING.PR.10] Require an ELB application or classic load balancer to have logging activated (p. 792)
- [CT.ELASTICLOADBALANCING.PR.11] Require any ELB classic load balancer to have connection draining activated (p. 800)
- [CT.ELASTICLOADBALANCING.PR.12] Require any ELB classic load balancer SSL/HTTPS listener to have a predefined security policy with a strong configuration (p. 805)
- [CT.ELASTICLOADBALANCING.PR.13] Require any ELB classic load balancer to have cross-zone load balancing activated (p. 811)
- [CT.ELASTICLOADBALANCING.PR.14] Require a Network Load Balancer to have cross-zone load balancing activated (p. 817)
- [CT.ELASTICLOADBALANCING.PR.15] Require that an Elastic Load Balancing v2 target group does not explicitly disable cross-zone load balancing (p. 822)

[CT.ELASTICLOADBALANCING.PR.1] Require any application load balancer listener default actions to redirect all HTTP requests to HTTPS

This control checks whether HTTP to HTTPS redirection is configured as a default action on HTTP listeners of Application Load Balancers.

- **Control objective:** Encrypt data in transit
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElasticLoadBalancingV2::Listener
- **AWS CloudFormation guard rule:** [CT.ELASTICLOADBALANCING.PR.1 rule specification (p. 736)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICLOADBALANCING.PR.1 rule specification (p. 736)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICLOADBALANCING.PR.1 example templates (p. 739)]

**Explanation**

Before you start to use your Application Load Balancer, you must add one or more listeners. A listener is a process that uses the configured protocol and port to check for connection requests. Listeners support HTTP and HTTPS protocols. You can use an HTTPS listener to offload the work of encryption and decryption to your Application Load Balancer. You should utilize redirect actions with Application Load Balancer to redirect any client HTTP request to an HTTPS request on port 443, to enforce encryption in transit.

**Usage considerations**

- This control evaluates only the default actions on Application Load Balancer listeners.

**Remediation for rule failure**

Configure a default HTTPS redirect action on Application Load Balancer HTTP listeners.

The examples that follow show how to implement this remediation.
Application Load Balancer Listener - Example

Application load balancer listener configured with a default action that redirects HTTP requests on port 80 to HTTPS requests on port 443, retaining the original host name, path, and query string. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "Listener": {
    "Type": "AWS::ElasticLoadBalancingV2::Listener",
    "Properties": {
      "LoadBalancerArn": {
        "Ref": "ApplicationLoadBalancer"
      },
      "Port": 80,
      "Protocol": "HTTP",
      "DefaultActions": [
        {
          "Type": "redirect",
          "RedirectConfig": {
            "Protocol": "HTTPS",
            "Port": 443,
            "Host": "#{host}",
            "Path": "/#{path}",
            "Query": "#{query}",
            "StatusCode": "HTTP_301"
          }
        }
      ]
    }
  }
}
```

**YAML example**

```yaml
Listener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    LoadBalancerArn: !Ref 'ApplicationLoadBalancer'
    Port: 80
    Protocol: HTTP
    DefaultActions:
      - Type: redirect
        RedirectConfig:
          Protocol: HTTPS
          Port: 443
          Host: "#{host}",
          Path: "/#{path}",
          Query: "#{query}",
          StatusCode: HTTP_301
```

CT.ELASTICLOADBALANCING.PR.1 rule specification

```plaintext
# ###################################################################
##       Rule Specification        ##
```

736
# Rule Identifier:
# alb_http_to_https_redirection_check
#
# Description:
# This control checks whether HTTP to HTTPS redirection is configured as a default action on HTTP listeners of Application Load Balancers.
#
# Reports on:
# AWS::ElasticLoadBalancingV2::Listener
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain any ElasticLoadBalancingV2 listener resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancingV2 listener
# And: 'Protocol' is set to a value other than 'HTTP'
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancingV2 listener
# And: 'Protocol' is set to 'HTTP'
# And: 'DefaultActions' is missing or is provided and an empty list
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancingV2 listener
# And: 'Protocol' is set to 'HTTP'
# And: 'DefaultActions' contains an action with 'Type' set to a value other than 'redirect'
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancingV2 listener
# And: 'Protocol' is set to 'HTTP'
# And: 'DefaultActions' contains an action with 'Type' set to a value of 'redirect'
# And: 'RedirectConfig.Protocol' is missing or set to a value other than 'HTTPS'
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancingV2 listener
# And: 'Protocol' is set to 'HTTP'
# And: All 'DefaultActions' have an action with 'Type' set to a value of 'redirect' and
#       'Protocol.RedirectConfig' set to the value 'HTTPS'
# Then: PASS
#
# Constants
let ELASTIC_LOAD_BALANCER_V2_LISTENER_TYPE = "AWS::ElasticLoadBalancingV2::Listener"
let INPUT_DOCUMENT = this

# Assignments
#
let elb_v2_listeners = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_V2_LISTENER_TYPE ]

# Primary Rules
#
rule alb_http_to_https_redirection_check when is_cfn_template(%INPUT_DOCUMENT)
%elb_v2_listeners not empty {
  check(%elb_v2_listeners.Properties)
  <<<
  [CT.ELASTICLOADBALANCING.PR.1]: Require any application load balancer listener
default actions to redirect all HTTP requests to HTTPS
  [FIX]: Configure a default HTTPS redirect action on application load balancer HTTP
  listeners.
  >>>
}

rule alb_http_to_https_redirection_check when is_cfn_hook(%INPUT_DOCUMENT,
%ELASTIC_LOAD_BALANCER_V2_LISTENER_TYPE) {
  check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_V2_LISTENER_TYPE.resourceProperties)
  <<<
  [CT.ELASTICLOADBALANCING.PR.1]: Require any application load balancer listener
default actions to redirect all HTTP requests to HTTPS
  [FIX]: Configure a default HTTPS redirect action on application load balancer HTTP
  listeners.
  >>>
}

# Parameterized Rules
#
rule check(elbv2_listener) {
  %elbv2_listener [

  # Scenario 2
  Protocol in [ "HTTP" ]

  # Scenarios 3
  DefaultActions exists
  DefaultActions is_list
  DefaultActions not empty

  # Scenario 4 and 5
  DefaultActions[*] {
    Type == "redirect"
    RedirectConfig exists
    RedirectConfig is_struct
    RedirectConfig {
      Protocol exists
      Protocol == "HTTPS"
    }
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
}
CT.ELASTICLOADBALANCING.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: ''
    SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
        - 1
        - Fn::GetAZs: ''
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
      - Ref: SubnetOne
      - Ref: SubnetTwo
    IpAddressType: ipv4
Listener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    LoadBalancerArn:
      Ref: ApplicationLoadBalancer
    Port: 80
    Protocol: HTTP
    DefaultActions:
      - Type: redirect
        RedirectConfig:
          Protocol: HTTPS
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: '
SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
        - 1
        - Fn::GetAZs: '
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
      - Ref: SubnetOne
      - Ref: SubnetTwo
    IpAddressType: ipv4
Listener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    LoadBalancerArn:
      Ref: ApplicationLoadBalancer
    Port: 80
    Protocol: HTTP
    DefaultActions:
      - Type: redirect
        RedirectConfig:
          Protocol: HTTP
          Port: 8080
          Host: "#{host}"
          Path: "#{path}"
          Query: "#{query}"
[CT.ELASTICLOADBALANCING.PR.2] Require any Amazon ELB application or network load balancer to have an AWS Certificate Manager certificate

This control checks whether your Elastic Load Balancing (ELB) application and network load balancers use certificates provided by AWS Certificate Manager (ACM).

- **Control objective:** Encrypt data in transit
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElasticLoadBalancingV2::Listener, AWS::ElasticLoadBalancingV2::ListenerCertificate
- **AWS CloudFormation guard rule:** [CT.ELASTICLOADBALANCING.PR.2 rule specification (p. 743)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICLOADBALANCING.PR.2 rule specification (p. 743)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICLOADBALANCING.PR.2 example templates (p. 746)]

**Explanation**

To create a certificate, use AWS Certificate Manager (ACM) or another tool that supports the SSL and TLS protocols, such as OpenSSL. AWS Control Tower recommends that you use AWS Certificate Manager to create or import certificates for your load balancer.

AWS Certificate Manager integrates with Amazon ELB application load balancers and network load balancers, so that you can deploy the certificate on your load balancer. We also recommend that you automatically renew these certificates.

**Usage considerations**

- This control applies only to HTTPS and TLS Amazon ELB listeners and ELB listener certificate resources that have one or more certificates configured.

**Remediation for rule failure**

Configure the Certificates property to use certificates provided by AWS Certificate Manager.

The examples that follow show how to implement this remediation.

**Amazon ELB Listener - Example**

Amazon ELB HTTPS listener configured with an AWS Certificate Manager SSL certificate. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ELBListener": {
    "Type": "AWS::ElasticLoadBalancingV2::Listener",
    "Properties": {
      "DefaultActions": [
        {
          "Type": "forward",
          "TargetGroupArn": {
```

741
YAML example

ELBLListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref 'TargetGroup'
        LoadBalancerArn: !Ref 'ApplicationLoadBalancer'
        Protocol: HTTPS
        Certificates:
          - CertificateArn: !Ref 'ACMCertificate'
            Port: 443

The examples that follow show how to implement this remediation.

Amazon ELB Listener Certificate - Example

Amazon ELB listener certificate configured with an AWS Certificate Manager SSL certificate. The example is shown in JSON and in YAML.

JSON example

```json
{
  "ELBListenerCertificate": {
    "Type": "AWS::ElasticLoadBalancingV2::ListenerCertificate",
    "Properties": {
      "ListenerArn": {
        "Ref": "Listener"
      },
      "Certificates": [
        { "CertificateArn": {
          "Ref": "ACMCertificate"
        } }
      ]
    }
  }
}
```
YAML example

```yaml
ELBListenerCertificate:
  Type: AWS::ElasticLoadBalancingV2::ListenerCertificate
  Properties:
    ListenerArn: !Ref 'Listener'
    Certificates:
      - CertificateArn: !Ref 'ACMCertificate'
```

CT.ELASTICLOADBALANCING.PR.2 rule specification

```plaintext
# ###########################################################################
##       Rule Specification        
# ###########################################################################
# # Rule Identifier:
#   elbv2_acm_certificate_required_check
# # Description:
#   This control checks whether your Elastic Load Balancing (ELB) application and network
load balancers use certificates provided by AWS Certificate Manager (ACM).
# # Reports on:
#   AWS::ElasticLoadBalancingV2::Listener, AWS::ElasticLoadBalancingV2::ListenerCertificate
# # Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# # Rule Parameters:
#   None
# # Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any ElasticLoadBalancingV2 listener or
#          listener certificate resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElasticLoadBalancingV2 listener resource
#     And: 'Protocol' is set to a value other than 'HTTPS' or 'TLS'
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElasticLoadBalancingV2 listener certificate
#          resource
#     And: 'Certificates' has not been provided or has been provided as an empty list
#     Then: SKIP
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an ElasticLoadBalancingV2 listener resource
```
# Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ElasticLoadBalancingV2 listener resource
And: 'Protocol' is set to 'HTTPS' or 'TLS'
And: One or more items in 'Certificates' do not match an ACM certificate ARN
Then: FAIL
# Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ElasticLoadBalancingV2 listener certificate resource
And: One or more items in 'Certificates' do not match an ACM certificate ARN
Then: FAIL
# Scenario: 7
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ElasticLoadBalancingV2 listener resource
And: 'Protocol' is set to 'HTTPS' or 'TLS'
And: All items in 'Certificates' match an ACM certificate ARN
Then: PASS
# Scenario: 8
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ElasticLoadBalancingV2 listener certificate resource
And: All items in 'Certificates' match an ACM certificate ARN
Then: PASS

# Constants
let ELASTIC_LOAD_BALANCER_V2_LISTENER_TYPE = "AWS::ElasticLoadBalancingV2::Listener"
let ELASTIC_LOAD_BALANCER_V2_CERTIFICATE_TYPE = "AWS::ElasticLoadBalancingV2::ListenerCertificate"
let ACM_CERTIFICATE_ARN_PATTERN = /arn:aws\[a-z0-9\-]*:acm:[a-z0-9\-]+:\d{12}:certificate\/[\w\-\d{1,64}]\[/
let INPUT_DOCUMENT = this

# Assignments
let elb_v2_listeners = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_V2_LISTENER_TYPE ]
let elb_v2_certificates = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_V2_CERTIFICATE_TYPE ]

# Primary Rules
rule elbv2_acm_certificate_required_check when is_cfn_template(%INPUT_DOCUMENT)
  %elb_v2_listeners not empty {
    check_listener(%elb_v2_listeners.Properties)
    <<
    [CT.ELASTICLOADBALANCING.PR.2]: Require any Amazon ELB application or network load balancer to have an AWS Certificate Manager certificate
    [FIX]: Configure the 'Certificates' property to use certificates provided by AWS Certificate Manager.
    >>
  }
rule elbv2_acm_certificate_required_check when is_cfn_hook(%INPUT_DOCUMENT,
  %ELASTIC_LOAD_BALANCER_V2_LISTENER_TYPE) {
    check_listener(%INPUT_DOCUMENT.
    %ELASTIC_LOAD_BALANCER_V2_LISTENER_TYPE.resourceProperties)
Proactive controls

[CT.ELASTICLOADBALANCING.PR.2]: Require any Amazon ELB application or network load balancer to have an AWS Certificate Manager certificate

[Fix]: Configure the 'Certificates' property to use certificates provided by AWS Certificate Manager.

rule elbv2_acm_certificate_required_check when is_cfn_template(%INPUT_DOCUMENT)
%elb_v2_certificates not empty {
  check_elbv2_listener_certificate(%elb_v2_certificates.Properties)
}

rule elbv2_acm_certificate_required_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_V2_CERTIFICATE_TYPE) {
  check_elbv2_listener_certificate(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_V2_CERTIFICATE_TYPE.resourceProperties)
}

# Parameterized Rules

rule check_listener(elbv2_listener) {
  %elbv2_listener[
    Protocol in ["HTTPS", "TLS"]
  ] {
    Certificates exists
    Certificates is_list
    Certificates not empty
    Certificates[*] {
      CertificateArn exists
      check_is_acm_certificate(CertificateArn)
    }
  }
}

rule check_elbv2_listener_certificate(listener_certificate) {
  %listener_certificate[
    Certificates exists
    Certificates is_list
    Certificates not empty
  ] {
    Certificates[*] {
      CertificateArn exists
      check_is_acm_certificate(CertificateArn)
    }
  }
}

rule check_is_acm_certificate(certificate) {
  %certificate {
You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

**Resources:**

- **VPC:**
  - Type: AWS::EC2::VPC
  - Properties:
    - CidrBlock: 10.0.0.0/16
    - EnableDnsSupport: 'true'
    - EnableDnsHostnames: 'true'

- **SubnetOne:**
  - Type: AWS::EC2::Subnet
  - Properties:
    - VpcId:
      - Ref: VPC
    - CidrBlock: 10.0.0.0/24
    - AvailabilityZone:
      - Fn::Select:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
- VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
- SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
VpcId:
  Ref: VPC
CidrBlock: 10.0.0.0/24
AvailabilityZone:
  Fn::Select:
  - 0
  - Fn::GetAZs: ''
SubnetTwo:
  Type: AWS::EC2::Subnet
Properties:
  VpcId:
    Ref: VPC
  CidrBlock: 10.0.1.0/24
  AvailabilityZone:
    Fn::Select:
    - 1
    - Fn::GetAZs: ''
TargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
Properties:
  Protocol: HTTP
  Port: 80
  VpcId:
    Ref: VPC
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
Properties:
  Scheme: internal
  Subnets:
    - Ref: SubnetOne
    - Ref: SubnetTwo
  IpAddressType: ipv4
Listener:
  Type: AWS::ElasticLoadBalancingV2::Listener
Properties:
  DefaultActions:
  - Type: forward
    TargetGroupArn:
      Ref: TargetGroup
    LoadBalancerArn:
      Ref: ApplicationLoadBalancer
  Protocol: HTTPS
  Certificates:
    Port: 443

[CT.ELASTICLOADBALANCING.PR.3] Require any application load balancer to have defensive or strictest desync mitigation mode activated

This control checks to ensure that an Application Load Balancer is configured with defensive or strictest desync mitigation mode.

- **Control objective:** Protect data integrity
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElasticLoadBalancingV2::LoadBalancer
- **AWS CloudFormation guard rule:** [CT.ELASTICLOADBALANCING.PR.3 rule specification (p. 750)]

Details and examples
• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICLOADBALANCING.PR.3 rule specification (p. 750)](#)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICLOADBALANCING.PR.3 example templates (p. 753)](#)

**Explanation**

HTTP desynchronization (desync) issues can lead to request smuggling and make applications vulnerable to request queue or cache poisoning. In turn, these vulnerabilities can lead to credential stuffing or execution of unauthorized commands. When configured with defensive or strictest desync mitigation mode, Application Load Balancers can protect your application from security issues that may be caused by HTTP desync.

**Remediation for rule failure**

Omit the load balancer attribute `routing.http.desync_mitigation_mode` or set the attribute to one of `defensive` or `strictest`.

The examples that follow show how to implement this remediation.

**Application Load Balancer - Example**

Application Load Balancer configured with defensive desync mitigation mode, by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ApplicationLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
      "Scheme": "internal",
      "Subnets": [
        {
          "Ref": "SubnetOne"
        },
        {
          "Ref": "SubnetTwo"
        }
      ],
      "IpAddressType": "ipv4",
      "Type": "application"
    }
  }
}
```

**YAML example**

```yaml
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
      - !Ref 'SubnetOne'
      - !Ref 'SubnetTwo'
    IpAddressType: ipv4
    Type: application
```
The examples that follow show how to implement this remediation.

**Application Load Balancer - Example**

Application Load Balancer configured with **strictest** desync mitigation mode, by means of the `routing.http.desync_mitigation_mode` load balancer attribute. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "ApplicationLoadBalancer": {
      "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
      "Properties": {
         "Scheme": "internal",
         "Subnets": [
            {
               "Ref": "SubnetOne"
            },
            {
               "Ref": "SubnetTwo"
            }
         ],
         "IpAddressType": "ipv4",
         "Type": "application",
         "LoadBalancerAttributes": [
            {
               "Key": "routing.http.desync_mitigation_mode",
               "Value": "strictest"
            }
         ]
      }
   }
}
```

**YAML example**

```yaml
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
      - !Ref 'SubnetOne'
      - !Ref 'SubnetTwo'
    IpAddressType: ipv4
    Type: application
    LoadBalancerAttributes:
      - Key: routing.http.desync_mitigation_mode
        Value: strictest
```

**CT.ELASTICLOADBALANCING.PR.3 rule specification**

```yaml
# ###################################################################
```
Proactive controls

### Rule Specification

#### Rule Identifier:

- alb_desync_mode_check

#### Description:

This control checks to ensure that an Application Load Balancer is configured with 'defensive' or 'strictest' desync mitigation mode.

#### Reports on:

- AWS::ElasticLoadBalancingV2::LoadBalancer

#### Evaluates:

- AWS CloudFormation, AWS CloudFormation hook

#### Rule Parameters:

- None

#### Scenarios:

1. **Scenario: 1**
   - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   - And: The input document does not contain any ELBv2 load balancer resources
   - Then: SKIP

2. **Scenario: 2**
   - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   - And: The input document contains an ELBv2 load balancer resource
   - And: 'Type' is set to a value other than 'application'
   - Then: SKIP

3. **Scenario: 3**
   - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   - And: The input document contains an ELBv2 load balancer resource
   - And: 'Type' is set to 'application' for the ELBv2 load balancer resource
   - And: 'LoadBalancerAttributes' have been specified on the ELBv2 load balancer resource
   - And: The 'LoadBalancerAttribute' 'routing.http.desync_mitigation_mode' has been provided
   - And: It is not one of 'defensive' or 'strictest'
   - Then: FAIL

4. **Scenario: 4**
   - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   - And: The input document contains an ELBv2 load balancer resource
   - And: 'Type' is set to 'application' for the ELBv2 load balancer resource
   - And: 'LoadBalancerAttributes' have not been specified on the ELBv2 load balancer resource or specified
   - as an empty list
   - Then: PASS

5. **Scenario: 5**
   - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   - And: The input document contains an ELBv2 load balancer resource
   - And: 'Type' is set to 'application' for the ELBv2 load balancer resource
   - And: 'LoadBalancerAttributes' have been specified on the ELBv2 load balancer resource
   - And: 'routing.http.desync_mitigation_mode' has not been provided as a 'LoadBalancerAttribute'
   - Then: PASS

6. **Scenario: 6**
   - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   - And: The input document contains an ELBv2 load balancer resource
   - And: 'Type' is set to 'application' for the ELBv2 load balancer resource

---
And: 'LoadBalancerAttributes' have been specified on the ELBv2 load balancer resource
And: The 'LoadBalancerAttribute' 'routing.http.desync_mitigation_mode' has been provided
 and is one of 'defensive' or 'strictest'
 Then: PASS

# Constants

let ELASTIC_LOAD_BALANCER_V2_TYPE = "AWS::ElasticLoadBalancingV2::LoadBalancer"
let ALLOWED_DESYNC_MODES = [ "defensive", "strictest" ]
let INPUT_DOCUMENT = this

# Assignments

let elastic_load_balancers = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_V2_TYPE ]

# Primary Rules

rule alb_desync_mode_check when is_cfn_template(%INPUT_DOCUMENT)
 %elastic_load_balancers not empty {
    check(%elastic_load_balancers.Properties)
    <<
    [CT.ELASTICLOADBALANCING.PR.3]: Require any application load balancer to have defensive or strictest desync mitigation mode activated
    [FIX]: Omit the load balancer attribute 'routing.http.desync_mitigation_mode' or set the attribute to one of 'defensive' or 'strictest'.
    >>
}

rule alb_desync_mode_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_V2_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_V2_TYPE.resourceProperties)
    <<
    [CT.ELASTICLOADBALANCING.PR.3]: Require any application load balancer to have defensive or strictest desync mitigation mode activated
    [FIX]: Omit the load balancer attribute 'routing.http.desync_mitigation_mode' or set the attribute to one of 'defensive' or 'strictest'.
    >>
}

# Parameterized Rules

rule check(elastic_load_balancer) {
    %elastic_load_balancer[
        # Scenario 2
        Type == "application"
    ] {
        # Scenario 4
        LoadBalancerAttributes not exists or
        check_application_load_balancer_attributes(this)
    }
}

rule check_application_load_balancer_attributes(application_load_balancer) {
    %application_load_balancer {
        LoadBalancerAttributes is_list
        LoadBalancerAttributes[
            # Scenario 5
Key exists
Key == "routing.http.desync_mitigation_mode"
] {
  # Scenarios 3 and 6
  Value exists
  Value in %ALLOWED_DESYNC_MODES
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.ELASTICLOADBALANCING.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
  SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/24
      AvailabilityZone:
        Fn::Select:
        - 0
        - Fn::GetAZs: ''
  SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.1.0/24
      AvailabilityZone:
        Fn::Select:
        - 1
        - Fn::GetAZs: ''
  ApplicationLoadBalancer:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
    Properties:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
  CidrBlock: 10.0.0.0/16
  EnableDnsSupport: 'true'
  EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
  VpcId:
    Ref: VPC
  CidrBlock: 10.0.0.0/24
  AvailabilityZone:
   Fn::Select:
    - 0
    - Fn::GetAZs: ''
SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
  VpcId:
    Ref: VPC
  CidrBlock: 10.0.1.0/24
  AvailabilityZone:
    Fn::Select:
    - 1
    - Fn::GetAZs: ''
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
  Scheme: internal
  Subnets:
  - Ref: SubnetOne
  - Ref: SubnetTwo
  IpAddressType: ipv4
  Type: application
  LoadBalancerAttributes:
  - Key: routing.http.desync_mitigation_mode
    Value: strictest

[CT.ELASTICLOADBALANCING.PR.4] Require that any application load balancer must be configured to drop HTTP headers

This control checks whether Application Load Balancers are configured to drop non-valid HTTP headers.

- **Control objective:** Protect configurations
• **Implementation**: AWS CloudFormation Guard Rule
• **Control behavior**: Proactive
• **Resource types**: AWS::ElasticLoadBalancingV2::LoadBalancer
• **AWS CloudFormation guard rule**: [CT.ELASTICLOADBALANCING.PR.4 rule specification](p. 756)

Details and examples

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICLOADBALANCING.PR.4 rule specification](p. 756)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICLOADBALANCING.PR.4 example templates](p. 758)

Explanation

By default, Application Load Balancers are not configured to drop non-valid HTTP header values. Removing these header values prevents HTTP desync attacks.

**Remediation for rule failure**

Set the load balancer attribute `routing.http.drop_invalid_header_fields.enabled` to `true`.

The examples that follow show how to implement this remediation.

**Application Load Balancer - Example**

Application Load Balancer configured to drop non-valid HTTP headers. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ApplicationLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
      "Scheme": "internal",
      "Type": "application",
      "Subnets": [
        {
          "Ref": "SubnetOne"
        },
        {
          "Ref": "SubnetTwo"
        }
      ],
      "IpAddressType": "ipv4",
      "LoadBalancerAttributes": [
        {
          "Key": "routing.http.drop_invalid_header_fields.enabled",
          "Value": "true"
        }
      ]
    }
  }
}
```

**YAML example**

```yaml
- ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internal
    Type: application
    Subnets:
      - Ref: SubnetOne
      - Ref: SubnetTwo
    IpAddressType: ipv4
    LoadBalancerAttributes:
      - Key: routing.http.drop_invalid_header_fields.enabled
        Value: true
```
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
Properties:
  Scheme: internal
  Type: application
  Subnets:
    - !Ref 'SubnetOne'
    - !Ref 'SubnetTwo'
  IpAddressType: ipv4
  LoadBalancerAttributes:
    - Key: routing.http.drop_invalid_header_fields.enabled
      Value: 'true'

CT.ELASTICLOADBALANCING.PR.4 rule specification

# #....................................................
## Rule Specification   ##
#....................................................

# Rule Identifier:
#    alb_http_drop_invalid_header_enabled_check
#
# Description:
#    This control checks whether Application Load Balancers are configured to drop non-valid
#    HTTP headers.
#
# Reports on:
#    AWS::ElasticLoadBalancingV2::LoadBalancer
#
# Evaluates:
#    AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#    None
#
# Scenarios:
#    Scenario: 1
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#        document
#        And: The input document does not contain any ELBv2 load balancer resources
#        Then: SKIP
#    Scenario: 2
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#        document
#        And: The input document contains an ELBv2 load balancer resource
#        And: 'Type' is set to a value other than 'application'
#        Then: SKIP
#    Scenario: 3
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#        document
#        And: The input document contains an ELBv2 load balancer resource
#        And: 'Type' is set to 'application' for the ELBv2 load balancer resource
#        And: 'LoadBalancerAttributes' have not been specified on the ELBv2 load balancer
#        resource
#        Then: FAIL
#    Scenario: 4
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#        document
#      And: The input document contains an ELBv2 load balancer resource
#      And: 'Type' is set to 'application' for the ELBv2 load balancer resource

756
And: 'LoadBalancerAttributes' have been specified on the ELBv2 load balancer resource
And: 'routing.http.drop_invalid_header_fields.enabled' has not been provided as a 'LoadBalancerAttribute'
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ELBv2 load balancer resource
And: 'Type' is set to 'application' for the ELBv2 load balancer resource
And: 'LoadBalancerAttributes' have been specified on the ELBv2 load balancer resource
And: The 'LoadBalancerAttribute' 'routing.http.drop_invalid_header_fields.enabled' has been provided
and is set to bool(false) or string(false)
Then: FAIL

Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an ELBv2 load balancer resource
And: 'Type' is set to 'application' for the ELBv2 load balancer resource
And: 'LoadBalancerAttributes' have been specified on the ELBv2 load balancer resource
And: The 'LoadBalancerAttribute' 'routing.http.drop_invalid_header_fields.enabled' has been provided and
is set to bool(true) or string(true)
Then: PASS

Constants
let ELASTIC_LOAD_BALANCER_V2_TYPE = "AWS::ElasticLoadBalancingV2::LoadBalancer"
let INPUT_DOCUMENT = this

Assignments
let elastic_load_balancers = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_V2_TYPE ]

Primary Rules
rule alb_http_drop_invalid_header_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%elastic_load_balancers not empty {
    check(%elastic_load_balancers.Properties)
    %elastic_load_balancers not empty {
        check(%elastic_load_balancers.Properties)
        <![CT.ELASTICLOADBALANCING.PR.4]: Require that any application load balancer must be configured to drop HTTP headers
        [FIX]: Set the load balancer attribute 'routing.http.drop_invalid_header_fields.enabled' to 'true'.
    }
}

rule alb_http_drop_invalid_header_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_V2_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_V2_TYPE.resourceProperties)
    <![CT.ELASTICLOADBALANCING.PR.4]: Require that any application load balancer must be configured to drop HTTP headers
    [FIX]: Set the load balancer attribute 'routing.http.drop_invalid_header_fields.enabled' to 'true'.
}

757
# Parameterized Rules

rule check(elastic_load_balancer) {
    %elastic_load_balancer[ Type == "application" ] {
        # Scenario 2
        LoadBalancerAttributes exists
        LoadBalancerAttributes is_list
        LoadBalancerAttributes not empty

        # Scenario 3, 4 and 5
        some LoadBalancerAttributes[*] {
            Key exists
            Value exists

            Key == "routing.http.drop_invalid_header_fields.enabled"
            Value in [ true, "true" ]
        }
    }
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.ELASTICLOADBALANCING.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
    Type: AWS::EC2::VPC
    Properties:
        CidrBlock: 10.0.0.0/16
        EnableDnsSupport: 'true'
        EnableDnsHostnames: 'true'

SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
        VpcId:
            Ref: VPC
        CidrBlock: 10.0.0.0/24
        AvailabilityZone:
            Fn::Select:
                - 0
                - Fn::GetAZs:

SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:

VPC:
  Type: AWS::EC2::VPC
  Properties:
   CidrBlock: 10.0.0.0/16
   EnableDnsSupport: 'true'
   EnableDnsHostnames: 'true'

SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
   VpcId:
    Ref: VPC
   CidrBlock: 10.0.0.0/24
   AvailabilityZone:
    Fn::Select:
     - 0
     - Fn::GetAZs: ''

SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
   VpcId:
    Ref: VPC
   CidrBlock: 10.0.1.0/24
   AvailabilityZone:
    Fn::Select:
     - 1
     - Fn::GetAZs: ''

ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
   Scheme: internal
   Type: application
   Subnets:
    - Ref: SubnetOne
    - Ref: SubnetTwo
   IpAddressType: ipv4
   LoadBalancerAttributes:
    - Key: routing.http.drop_invalid_header_fields.enabled
      Value: "false"
[CT.ELASTICLOADBALANCING.PR.5] Require that application load balancer deletion protection is activated

Checks whether Elastic Load Balancing (ELB) has deletion protection activated.

- **Control objective**: Improve availability
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::ElasticLoadBalancingV2::LoadBalancer
- **AWS CloudFormation guard rule**: CT.ELASTICLOADBALANCING.PR.5 rule specification (p. 761)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ELASTICLOADBALANCING.PR.5 rule specification (p. 761)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.ELASTICLOADBALANCING.PR.5 example templates (p. 763)

**Explanation**

Activate deletion protection to protect your Application Load Balancer from deletion.

**Remediation for rule failure**

Set the load balancer attribute `deletion_protection.enabled` to `true`.

The examples that follow show how to implement this remediation.

**Application Load Balancer - Example**

Application Load Balancer configured with deletion protection active. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "Elb": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
      "Scheme": "internal",
      "Type": "application",
      "Subnets": [
        {
          "Ref": "SubnetOne"
        },
        {
          "Ref": "SubnetTwo"
        }
      ],
      "IpAddressType": "ipv4",
      "LoadBalancerAttributes": [
        {
          "Key": "deletion_protection.enabled",
          "Value": "true"
        }
      ]
  }
}
```

760
YAML example

Elb:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
  
  Scheme: internal
  Type: application
  Subnets:
  - !Ref 'SubnetOne'
  - !Ref 'SubnetTwo'
  IpAddressType: ipv4
  LoadBalancerAttributes:
    - Key: deletion_protection.enabled
      Value: 'true'

CT.ELASTICLOADBALANCING.PR.5 rule specification

# # Rule Specification #
# # Rule Identifier: #
# # elbv2_deletion_protection_enabled_check #
# # Description: #
# Checks whether Elastic Load Balancing (ELB) has deletion protection activated. #
# # Reports on: #
# AWS::ElasticLoadBalancingV2::LoadBalancer #
# # Evaluates: #
# AWS CloudFormation, AWS CloudFormation hook #
# # Rule Parameters: #
# None #
# # Scenarios: #
# Scenario: 1 #
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document #
# And: The input document does not contain any ELBv2 LoadBalancer resource #
# Then: SKIP #
# Scenario: 2 #
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document #
# And: The input document contains an ELBv2 LoadBalancer resource #
# And: 'LoadBalancerAttributes' have not been specified or is an empty list on the ELBv2 resource #
# Then: FAIL #
# Scenario: 3 #
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document #
# And: The input document contains an ELBv2 LoadBalancer resource #
# And: 'LoadBalancerAttributes' have been specified on the ELBv2 LoadBalancer resource
And: 'deletion_protection.enabled' has not been provided as a
'LoadBalancerAttribute'
  Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an ELBv2 LoadBalancer resource
# And: 'LoadBalancerAttributes' have been specified on the ELBv2 LoadBalancer
resource
# And: The 'LoadBalancerAttribute' 'deletion_protection.enabled' has been provided
and is set to bool(false) or
  string(false)
  Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an ELBv2 LoadBalancer Resource
# And: 'LoadBalancerAttributes' have been specified on the ELBv2 LoadBalancer
resource
# And: The 'LoadBalancerAttribute' 'deletion_protection.enabled' has been provided
and is set to bool(true) or
  string(true)
  Then: PASS
#
# Constants
#
let ELASTIC_LOAD_BALANCER_V2_TYPE = "AWS::ElasticLoadBalancingV2::LoadBalancer"
let INPUT_DOCUMENT = this
#
# Assignments
#
let elastic_load_balancers = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_V2_TYPE ]
#
# Primary Rules
#
rule elbv2_deletion_protection_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
  %elastic_load_balancers not empty {
    check(%elastic_load_balancers.Properties)
    <<
    [CT.ELASTICLOADBALANCING.PR.5]: Require that application load balancer deletion
    protection is activated
    [FIX]: Set the load balancer attribute 'deletion_protection.enabled' to 'true'.
    >>
} rule elbv2_deletion_protection_enabled_check when is_cfn_hook(%INPUT_DOCUMENT,
  %ELASTIC_LOAD_BALANCER_V2_TYPE) {
  check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_V2_TYPE.resourceProperties)
  <<
  [CT.ELASTICLOADBALANCING.PR.5]: Require that application load balancer deletion
  protection is activated
  [FIX]: Set the load balancer attribute 'deletion_protection.enabled' to 'true'.
  >>
}
#
# Parameterized Rules
#
rule check(elastic_load_balancer) {
  %elastic_load_balancer {
    # Scenario 2
    LoadBalancerAttributes exists
    LoadBalancerAttributes is_list
  }
Proactive controls

LoadBalancerAttributes not empty

# Scenario 3, 4 and 5
some LoadBalancerAttributes[*] {
    Key exists
    Value exists
    Key == "deletion_protection.enabled"
    Value in [ true, "true" ]
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.ELASTICLOADBALANCING.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
    Type: AWS::EC2::VPC
    Properties:
        CidrBlock: 10.0.0.0/16
        EnableDnsSupport: 'true'
        EnableDnsHostnames: 'true'
SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
        VpcId:
            Ref: VPC
        CidrBlock: 10.0.0.0/24
        AvailabilityZone:
            Fn::Select:
                - 0
                - Fn::GetAZs: '
SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
        VpcId:
            Ref: VPC
        CidrBlock: 10.0.1.0/24
        AvailabilityZone:
            Fn::Select:
                - 1
                - Fn::GetAZs: ''
ApplicationLoadBalancer:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

[CT.ELASTICLOADBALANCING.PR.6] Require that application and network load balancer access logging is activated

This control checks whether your Elastic Load Balancing (ELB) application and network load balancers have logging activated.
**Control objective:** Establish logging and monitoring

**Implementation:** AWS CloudFormation Guard Rule

**Control behavior:** Proactive

**Resource types:** AWS::ElasticLoadBalancingV2::LoadBalancer

**AWS CloudFormation guard rule:** CT.ELASTICLOADBALANCING.PR.6 rule specification (p. 767)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ELASTICLOADBALANCING.PR.6 rule specification (p. 767)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.ELASTICLOADBALANCING.PR.6 example templates (p. 770)

**Explanation**

Elastic Load Balancing provides access logs that capture detailed information about requests sent to your load balancer. Each log contains information such as the time the request was received, the client's IP address, latencies, request paths, and server responses. You can use these access logs to analyze traffic patterns and to troubleshoot issues.

**Usage considerations**

- This control applies only to ELB load balancer types of application and network.

**Remediation for rule failure**

Set the load balancer attribute access_logs.s3.enabled to true, and set access_logs.s3.bucket to reach an S3 bucket that’s configured to receive application load balancer or network load balancer access logs.

The examples that follow show how to implement this remediation.

**Application Load Balancer - Example**

Application Load Balancer configured with access logging activated. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ApplicationLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
      "Scheme": "internal",
      "Subnets": [
        
          { "Ref": "SubnetOne" }
        ,
        { "Ref": "SubnetTwo" }
      ],
      "IpAddressType": "ipv4",
      "Type": "application",
      "LoadBalancerAttributes": [
      
      ]
    }
  }
}
```
"Key": "access_logs.s3.enabled",
"Value": true
],
[
"Key": "access_logs.s3.bucket",
"Value": {
  "Ref": "LoggingBucket"
}
]
}
}

YAML example

```yaml
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
    - !Ref 'SubnetOne'
    - !Ref 'SubnetTwo'
    IpAddressType: ipv4
    Type: application
    LoadBalancerAttributes:
    - Key: access_logs.s3.enabled
      Value: true
    - Key: access_logs.s3.bucket
      Value: !Ref 'LoggingBucket'
```

The examples that follow show how to implement this remediation.

**Network Load Balancer - Example**

Network Load Balancer configured with access logging activated. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "NetworkLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
      "Scheme": "internal",
      "Subnets": [
        {
          "Ref": "SubnetOne"
        },
        {
          "Ref": "SubnetTwo"
        }
      ],
      "IpAddressType": "ipv4",
      "Type": "network",
      "LoadBalancerAttributes": [
        {
          "Key": "access_logs.s3.enabled",
```
"Value": true
}

NetworkLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
      - !Ref 'SubnetOne'
      - !Ref 'SubnetTwo'
    IpAddressType: ipv4
    Type: network
    LoadBalancerAttributes:
      - Key: access_logs.s3.enabled
        Value: true
      - Key: access_logs.s3.bucket
        Value: !Ref 'LoggingBucket'

CT.ELASTICLOADBALANCING.PR.6 rule specification

# ###################################################################
##       Rule Specification       ##
# ###################################################################
#
# Rule Identifier:
#  elbv2_logging_enabled_check
#
# Description:
#  This control checks whether your Elastic Load Balancing (ELB) application and network
#  load balancers have logging activated.
#
# Reports on:
#  AWS::ElasticLoadBalancingV2::LoadBalancer
#
# Evaluates:
#  AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#  None
#
# Scenarios:
#  Scenario: 1
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#         document
#    And: The input document does not contain any ElasticLoadBalancingV2 LoadBalancer
#         resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancingV2 LoadBalancer resource
# And: 'Type' is set to a value other than 'application' or 'network'
# Then: SKIP

# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancingV2 LoadBalancer resource
# And: The LoadBalancer is of type 'application' or 'network'
# And: 'LoadBalancerAttributes' has not been provided or is an empty list
# Then: FAIL

# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancingV2 LoadBalancer resource
# And: The LoadBalancer is of type 'application' or 'network'
# And: 'LoadBalancerAttributes' has not been provided or is an empty list
# Then: FAIL

# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancingV2 LoadBalancer resource
# And: The LoadBalancer is of type 'application' or 'network'
# And: 'LoadBalancerAttributes' has not been provided or is an empty list
# Then: FAIL

# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancingV2 LoadBalancer resource
# And: The LoadBalancer is of type 'application' or 'network'
# And: 'LoadBalancerAttributes' has not been provided or is an empty list
# Then: FAIL

# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancingV2 LoadBalancer resource
# And: The LoadBalancer is of type 'application' or 'network'
# And: 'LoadBalancerAttributes' has not been provided or is an empty list
# Then: FAIL

# Constants
let ELASTIC_LOAD_BALANCER_V2_TYPE = "AWS::ElasticLoadBalancingV2::LoadBalancer"
let INPUT_DOCUMENT = this

# Assignments
let elastic_load_balancers = Resources.[*| Type == %ELASTIC_LOAD_BALANCER_V2_TYPE ]

# Primary Rules
# Proactive controls

## Rule: elbv2_logging_enabled_check

```python
rule elbv2_logging_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
    %elastic_load_balancers not empty {
    check(%elastic_load_balancers.Properties)
    <<
    [CT.ELASTICLOADBALANCING.PR.6]: Require that application and network load balancer access logging is activated
    [FIX]: Set the load balancer attribute 'access_logs.s3.enabled' to 'true', and set 'access_logs.s3.bucket' to reach an Amazon S3 bucket that's configured to receive application load balancer or network load balancer access logs.
    >>
}
```

## Rule: elbv2_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_V2_TYPE)

```python
rule elbv2_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_V2_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_V2_TYPE.resourceProperties)
    <<
    [CT.ELASTICLOADBALANCING.PR.6]: Require that application and network load balancer access logging is activated
    [FIX]: Set the load balancer attribute 'access_logs.s3.enabled' to 'true', and set 'access_logs.s3.bucket' to reach an Amazon S3 bucket that's configured to receive application load balancer or network load balancer access logs.
    >>
}
```

## Parameterized Rules

### Rule: check(elastic_load_balancer) [{

```python
rule check(elastic_load_balancer[ Type in ["application", "network"] ] {
    %elastic_load_balancer[ Type in ["application", "network"] ] {
        # Scenario 3
        LoadBalancerAttributes exists
        LoadBalancerAttributes is_list
        LoadBalancerAttributes not empty

        # Scenario 4, 5, 6 and 7
        some LoadBalancerAttributes[*] {
            Key exists
            Value exists

            Key == "access_logs.s3.enabled"
            Value in [ true, "true" ]
        }

        some LoadBalancerAttributes[*] {
            Key exists
            Value exists

            Key == "access_logs.s3.bucket"
            check_is_string_and_not_empty(Value) or check_local_references(%INPUT_DOCUMENT, Value, "AWS::S3::Bucket")
        }
    }
}
```

## Utility Rules

### Rule: is_cfn_template(doc) [{

```python
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
```
AWS Control Tower User Guide
Proactive controls

rule is_cfn_hook(doc, RESOURCE_TYPE) {
%doc.%RESOURCE_TYPE.resourceProperties exists
}
rule check_is_string_and_not_empty(value) {
%value {
this is_string
this != /\A\s*\z/
}
}
rule check_local_references(doc, reference_properties, referenced_resource_type) {
%reference_properties {
'Fn::GetAtt' {
query_for_resource(%doc, this[0], %referenced_resource_type)
<<Local Stack reference was invalid>>
} or Ref {
query_for_resource(%doc, this, %referenced_resource_type)
<<Local Stack reference was invalid>>
}
}
}
rule query_for_resource(doc, resource_key, referenced_resource_type) {
let referenced_resource = %doc.Resources[ keys == %resource_key ]
%referenced_resource not empty
%referenced_resource {
Type == %referenced_resource_type
}
}

CT.ELASTICLOADBALANCING.PR.6 example templates
You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.
PASS Example - Use this template to verify a compliant resource creation.

Mappings:
RegionToELBAccountId:
us-east-1:
AccountId: '127311923021'
us-west-1:
AccountId: '027434742980'
us-west-2:
AccountId: '797873946194'
ca-central-1:
AccountId: '985666609251'
eu-west-1:
AccountId: '156460612806'
ap-northeast-1:
AccountId: '582318560864'
ap-northeast-2:
AccountId: '600734575887'
ap-southeast-1:
AccountId: '114774131450'
ap-southeast-2:
AccountId: '783225319266'
ap-south-1:
AccountId: '718504428378'
us-east-2:

770


AccountId: '033677994240'
sa-east-1:
  AccountId: '507241528517'
eu-central-1:
  AccountId: '054676820928'
af-south-1:
  AccountId: '098369216593'
ap-east-1:
  AccountId: '754344448648'
ap-southeast-3:
  AccountId: '589379963580'
ap-northeast-3:
  AccountId: '383597477331'
eu-west-2:
  AccountId: '652711504416'
eu-south-1:
  AccountId: '635631232127'
eu-west-3:
  AccountId: '009996457667'
eu-north-1:
  AccountId: '897822967062'
me-south-1:
  AccountId: '076674570225'
us-gov-west-1:
  AccountId: '048591011584'
us-gov-east-1:
  AccountId: '190560391635'
RegionToARNPrefix:
  us-east-1:
    ARNPrefix: 'arn:aws:'
  us-west-1:
    ARNPrefix: 'arn:aws:'
  us-west-2:
    ARNPrefix: 'arn:aws:'
  ca-central-1:
    ARNPrefix: 'arn:aws:'
eu-west-1:
    ARNPrefix: 'arn:aws:'
ap-northeast-1:
    ARNPrefix: 'arn:aws:'
ap-northeast-2:
    ARNPrefix: 'arn:aws:'
ap-southeast-1:
    ARNPrefix: 'arn:aws:'
ap-southeast-2:
    ARNPrefix: 'arn:aws:'
ap-south-1:
    ARNPrefix: 'arn:aws:'
  us-east-2:
    ARNPrefix: 'arn:aws:'
  sa-east-1:
    ARNPrefix: 'arn:aws:'
  eu-central-1:
    ARNPrefix: 'arn:aws:'
  af-south-1:
    ARNPrefix: 'arn:aws:'
ap-east-1:
    ARNPrefix: 'arn:aws:'
ap-southeast-3:
    ARNPrefix: 'arn:aws:'
ap-northeast-3:
    ARNPrefix: 'arn:aws:'
eu-west-2:
    ARNPrefix: 'arn:aws:'
eu-south-1:
    ARNPrefix: 'arn:aws:'
  eu-south-1:
    ARNPrefix: 'arn:aws:'
  eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-central-1:
    ARNPrefix: 'arn:aws:'
  af-south-1:
    ARNPrefix: 'arn:aws:'
ap-east-1:
    ARNPrefix: 'arn:aws:'
ap-southeast-3:
    ARNPrefix: 'arn:aws:'
ap-northeast-3:
    ARNPrefix: 'arn:aws:'
eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-south-1:
    ARNPrefix: 'arn:aws:'
  eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-central-1:
    ARNPrefix: 'arn:aws:'
  af-south-1:
    ARNPrefix: 'arn:aws:'
ap-east-1:
    ARNPrefix: 'arn:aws:'
ap-southeast-3:
    ARNPrefix: 'arn:aws:'
ap-northeast-3:
    ARNPrefix: 'arn:aws:'
eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-south-1:
    ARNPrefix: 'arn:aws:'
  eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-central-1:
    ARNPrefix: 'arn:aws:'
  af-south-1:
    ARNPrefix: 'arn:aws:'
ap-east-1:
    ARNPrefix: 'arn:aws:'
ap-southeast-3:
    ARNPrefix: 'arn:aws:'
ap-northeast-3:
    ARNPrefix: 'arn:aws:'
eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-south-1:
    ARNPrefix: 'arn:aws:'
  eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-central-1:
    ARNPrefix: 'arn:aws:'
  af-south-1:
    ARNPrefix: 'arn:aws:'
ap-east-1:
    ARNPrefix: 'arn:aws:'
ap-southeast-3:
    ARNPrefix: 'arn:aws:'
ap-northeast-3:
    ARNPrefix: 'arn:aws:'
eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-south-1:
    ARNPrefix: 'arn:aws:'
  eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-central-1:
    ARNPrefix: 'arn:aws:'
  af-south-1:
    ARNPrefix: 'arn:aws:'
ap-east-1:
    ARNPrefix: 'arn:aws:'
ap-southeast-3:
    ARNPrefix: 'arn:aws:'
ap-northeast-3:
    ARNPrefix: 'arn:aws:'
eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-south-1:
    ARNPrefix: 'arn:aws:'
  eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-central-1:
    ARNPrefix: 'arn:aws:'
  af-south-1:
    ARNPrefix: 'arn:aws:'
ap-east-1:
    ARNPrefix: 'arn:aws:'
ap-southeast-3:
    ARNPrefix: 'arn:aws:'
ap-northeast-3:
    ARNPrefix: 'arn:aws:'
eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-south-1:
    ARNPrefix: 'arn:aws:'
  eu-west-2:
    ARNPrefix: 'arn:aws:'
  eu-central-1:
    ARNPrefix: 'arn:aws:'
  af-south-1:
    ARNPrefix: 'arn:aws:'
ap-east-1:
    ARNPrefix: 'arn:aws:'
ap-southeast-3:
    ARNPrefix: 'arn:aws:'
ap-northeast-3:
eu-west-3:
  ARNPrefix: 'arn:aws:'
eu-north-1:
  ARNPrefix: 'arn:aws:'
me-south-1:
  ARNPrefix: 'arn:aws:'
us-gov-west-1:
  ARNPrefix: 'arn:aws-us-gov:'
us-gov-east-1:
  ARNPrefix: 'arn:aws-us-gov:'
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
  SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/24
      AvailabilityZone:
        Fn::Select:
          - 0
          - Fn::GetAZs: ''
  SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.1.0/24
      AvailabilityZone:
        Fn::Select:
          - 1
          - Fn::GetAZs: ''
  LoggingBucket:
    Type: AWS::S3::Bucket
  LoggingBucketPolicy:
    Type: AWS::S3::BucketPolicy
    Properties:
      Bucket:
        Ref: LoggingBucket
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Action:
            - 's3:PutObject'
          - Effect: Allow
          - Resource:
            Fn::Join:
            - ''
            - - Fn::FindInMap: [RegionToARNPrefix, !Ref 'AWS::Region', ARNPrefix]
              - 's3:::'
            - Ref: LoggingBucket
            - /AWSLogs/
            - Ref: AWS::AccountId
            - /*
      Principal:
        AWS:
          Fn::FindInMap: [RegionToELBAccountId, !Ref 'AWS::Region', AccountId]
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internal
Subnets:
- Ref: SubnetOne
- Ref: SubnetTwo
IpAddressType: ipv4
Type: application
LoadBalancerAttributes:
- Key: access_logs.s3.enabled
  Value: true
- Key: access_logs.s3.bucket
  Value: Ref: LoggingBucket

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
  SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/24
      AvailabilityZone:
        Fn::Select:
          - 0
          - Fn::GetAZs: '
  SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.1.0/24
      AvailabilityZone:
        Fn::Select:
          - 1
          - Fn::GetAZs: '
  ApplicationLoadBalancer:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
    Properties:
      Scheme: internal
      Subnets:
        - Ref: SubnetOne
        - Ref: SubnetTwo
      IpAddressType: ipv4
      Type: application

[CT.ELASTICLOADBALANCING.PR.7] Require any classic load balancer to have multiple Availability Zones configured

This control checks whether an Elastic Load Balancing (ELB) classic load balancer has been configured with multiple Availability Zones.

- **Control objective**: Improve availability
• **Implementation**: AWS CloudFormation Guard Rule
• **Control behavior**: Proactive
• **Resource types**: AWS::ElasticLoadBalancing::LoadBalancer
• **AWS CloudFormation guard rule**: CT.ELASTICLOADBALANCING.PR.7 rule specification (p. 777)

**Details and examples**

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ELASTICLOADBALANCING.PR.7 rule specification (p. 777)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.ELASTICLOADBALANCING.PR.7 example templates (p. 779)

**Explanation**

A Classic Load Balancer can be set up to distribute incoming requests across Amazon EC2 instances in a single Availability Zone or multiple Availability Zones. A Classic Load Balancer that does not span multiple Availability Zones is unable to redirect traffic to targets in another Availability Zone, in case the sole configured Availability Zone becomes unavailable.

**Remediation for rule failure**

Configure Classic Load Balancers with two or more subnets or Availability Zones.

The examples that follow show how to implement this remediation.

**Classic Load Balancer - Example One**

Classic Load Balancer configured with two Availability Zones. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ClassicLoadBalancer": {
        "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
        "Properties": {
            "Scheme": "internet-facing",
            "Listeners": [
                {
                    "InstancePort": "80",
                    "InstanceProtocol": "HTTP",
                    "LoadBalancerPort": "443",
                    "Protocol": "HTTPS",
                    "PolicyNames": [
                        "Sample-SSLNegotiation-Policy"
                    ],
                    "SSLCertificateId": {
                        "Ref": "ACMCertificate"
                    }
                }
            ],
            "Policies": [
                {
                    "PolicyName": "Sample-SSLNegotiation-Policy",
                    "PolicyType": "SSLNegotiationPolicyType",
                    "Attributes": [
                        {
                            "Name": "Reference-Security-Policy",
                            "Reference": ""
"Value": "ELBSecurityPolicy-TLS-1-2-2017-01"

YAML example

YAML example

ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internet-facing
    Listeners:
      - InstancePort: '80'
        InstanceProtocol: HTTP
        LoadBalancerPort: '443'
        Protocol: HTTPS
    PolicyNames:
      - Sample-SSLNegotiation-Policy
    SSLCertificateId: !Ref 'ACMCertificate'
  Policies:
    - PolicyName: Sample-SSLNegotiation-Policy
      PolicyType: SSLNegotiationPolicyType
      Attributes:
        - Name: Reference-Security-Policy
          Value: ELBSecurityPolicy-TLS-1-2-2017-01
  AvailabilityZones:
    - !Select
      - 0
      - !GetAZs ''
    - !Select
      - 1
      - !GetAZs ''

The examples that follow show how to implement this remediation.

**Classic Load Balancer - Example Two**

Classic Load Balancer configured with two subnets. The example is shown in JSON and in YAML.
JSON example

```
{
  "ClassicLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
    "Properties": {
      "Scheme": "internet-facing",
      "Listeners": [
        {
          "InstancePort": "80",
          "InstanceProtocol": "HTTP",
          "LoadBalancerPort": "443",
          "Protocol": "HTTPS",
          "PolicyNames": [
            "Sample-SSLNegotiation-Policy"
          ],
          "SSLCertificateId": {
            "Ref": "ACMCertificate"
          }
        }
      ],
      "Policies": [
        {
          "PolicyName": "Sample-SSLNegotiation-Policy",
          "PolicyType": "SSLNegotiationPolicyType",
          "Attributes": [
            {
              "Name": "Reference-Security-Policy",
              "Value": "ELBSecurityPolicy-TLS-1-2-2017-01"
            }
          ]
        }
      ],
      "AvailabilityZones": [
        {
          "Fn::Select": [0,
            {
              "Fn::GetAZs": ""
            }
          ]
        },
        {
          "Fn::Select": [1,
            {
              "Fn::GetAZs": ""
            }
          ]
        }
      ]
    }
  }
}
```

YAML example

```
ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internet-facing
```
Listeners:
- InstancePort: '80'
  InstanceProtocol: HTTP
- LoadBalancerPort: '443'
  Protocol: HTTPS
PolicyNames:
- Sample-SSLNegotiation-Policy
SSLCertificateId: !Ref 'ACMCertificate'
Policies:
- PolicyName: Sample-SSLNegotiation-Policy
  PolicyType: SSLNegotiationPolicyType
Attributes:
- Name: Reference-Security-Policy
  Value: ELBSecurityPolicy-TLS-1-2-2017-01
AvailabilityZones:
- !Select
  - 0
  - !GetAZs ''
- !Select
  - 1
  - !GetAZs ' 

CT.ELASTICLOADBALANCING.PR.7 rule specification

```bash
# ###################################################################
## Rule Specification ##
# ###################################################################

# Rule Identifier:
# elb_multiple_az_check

# Description:
# This control checks whether an Elastic Load Balancing (ELB) Classic Load Balancer has
# been configured with multiple Availability Zones.

# Reports on:
# AWS::ElasticLoadBalancing::LoadBalancer

# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain any Elastic Load Balancing load balancer resources
# Then: SKIP
# Scenario: 2
# Given: The input document contains an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Elastic Load Balancing load balancer resource
# And: Neither 'AvailabilityZones' or 'Subnets' have been specified
# Then: FAIL
# Scenario: 3
# Given: The input document contains an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Elastic Load Balancing load balancer resource
```
Proactive controls

# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Elastic Load Balancing load balancer resource
# And: 'Subnets' been specified on the Elastic Load Balancing load balancer resource
# And: The number of entries in 'Subnets' is < 2
# Then: FAIL

# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Elastic Load Balancing load balancer resource
# And: 'AvailabilityZones' been specified on the Elastic Load Balancing load balancer resource
# And: The number of entries in 'AvailabilityZones' is >= 2
# Then: PASS

# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Elastic Load Balancing load balancer resource
# And: 'Subnets' been specified on the Elastic Load Balancing load balancer resource
# And: The number of entries in 'Subnets' is >= 2
# Then: PASS

# Constants
#
let ELASTIC_LOAD_BALANCER_TYPE = "AWS::ElasticLoadBalancing::LoadBalancer"
let INPUT_DOCUMENT = this
#
# Assignments
#
let classic_load_balancers = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_TYPE ]
#
# Primary Rules
#
rule elb_multiple_az_check when is_cfn_template(%INPUT_DOCUMENT) {
  %classic_load_balancers not empty {
    check(%classic_load_balancers.Properties) <<
    [CT.ELASTICLOADBALANCING.PR.7]: Require any classic load balancer to have multiple Availability Zones configured
    [FIX]: Configure Classic Load Balancers with two or more subnets or Availability Zones.
    >>
  }
}
rule elb_multiple_az_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_TYPE) {
  check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_TYPE.resourceProperties) <<
  [CT.ELASTICLOADBALANCING.PR.7]: Require any classic load balancer to have multiple Availability Zones configured
  [FIX]: Configure Classic Load Balancers with two or more subnets or Availability Zones.
  >>
}
#
# Parameterized Rules

```perl
#
rule check(classic_load_balancer) {
  %classic_load_balancer {
    # Scenario 2
    AvailabilityZones exists or
    Subnets exists

    when AvailabilityZones exists {
      # Scenarios 3 and 5
      two_or_more_entries(AvailabilityZones)
      AvailabilityZones[0] not in AvailabilityZones[1]
    }

    when Subnets exists {
      # Scenarios 4 and 6
      two_or_more_entries(Subnets)
    }
  }
}
```

```perl
rule two_or_more_entries(list_property) {
  %list_property {
    this is_list
    this not empty
    this[0] exists
    this[1] exists
  }
}
```

# Utility Rules

```perl
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
```

```perl
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

CT.ELASTICLOADBALANCING.PR.7 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:

ACMCertificate:
  Type: "AWS::CertificateManager::Certificate"
Properties:
  DomainName: example.com
  ValidationMethod: DNS
  DomainValidationOptions:
    - DomainName: www.example.com
  HostedZoneId: ZZZHHHHWWWWAAA
ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
Properties:
    Scheme: internet-facing
    Listeners:
    - Protocol: HTTPS
      InstancePort: '80'
      InstanceProtocol: HTTP
      LoadBalancerPort: '443'
    PolicyNames:
    - Example-SSLNegotiation-Policy
    SSLCertificateId:
      Ref: ACMCertificate
    Policies:
    - PolicyName: Example-SSLNegotiation-Policy
      PolicyType: SSLNegotiationPolicyType
      Attributes:
      - Name: Reference-Security-Policy
      Value: ELBSecurityPolicy-TLS-1-2-2017-01
    AvailabilityZones:
    - Fn::Select:
    - 0
    - Fn::GetAZs: ''
    - Fn::Select:
    - 1
    - Fn::GetAZs: ''

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
    Type: AWS::EC2::VPC
    Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
Subnet:
    Type: AWS::EC2::Subnet
    Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
      - 0
      - Fn::GetAZs: ''
ClassicLoadBalancer:
    Type: AWS::ElasticLoadBalancing::LoadBalancer
    Properties:
    Scheme: internal
    Listeners:
    - Protocol: HTTP
      InstancePort: 80
      LoadBalancerPort: 80
    Subnets:
    - Ref: Subnet
[CT.ELASTICLOADBALANCING.PR.8] Require any classic load balancer SSL/HTTPS listener to have a certificate provided by AWS Certificate Manager

This control checks whether classic load balancers use HTTPS/SSL certificates provided by AWS Certificate Manager.

- **Control objective:** Encrypt data in transit
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElasticLoadBalancing::LoadBalancer
- **AWS CloudFormation guard rule:** CT.ELASTICLOADBALANCING.PR.8 rule specification (p. 782)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: 
  CT.ELASTICLOADBALANCING.PR.8 rule specification (p. 782)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: 
  CT.ELASTICLOADBALANCING.PR.8 example templates (p. 785)

**Explanation**

To create a certificate, you can use either ACM or a tool that supports the SSL and TLS protocols, such as OpenSSL. Security Hub recommends that you use ACM to create or import certificates for your load balancer.

ACM integrates with Classic Load Balancers, so that you can deploy the certificate on your load balancer. You also should renew these certificates automatically.

**Usage considerations**

- This control applies only to Classic Load Balancers configured with HTTPS or SSL listeners.

**Remediation for rule failure**

Configure Classic Load Balancers to use certificates provided by AWS Certificate Manager (ACM).

The examples that follow show how to implement this remediation.

**Classic Load Balancer - Example**

Classic Load Balancer configured with an HTTPS listener and AWS Certificate Manager SSL certificate. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ClassicLoadBalancer": {
        "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
        "Properties": {
            "Scheme": "internal",
            "Subnets": [
                { "Ref": "SubnetOne" },
                { "Ref": "SubnetTwo" }
            ]
        }
    }
}
```
Proactive controls

```
YAML example

ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
      - !Ref 'SubnetOne'
      - !Ref 'SubnetTwo'
    Policies:
      - PolicyName: Example-SSLNegotiation-Policy
        PolicyType: SSLNegotiationPolicyType
        Attributes:
          - Name: Reference-Security-Policy
          - Value: ELBSecurityPolicy-TLS-1-2-2017-01
    Listeners:
      - InstancePort: 80
        InstanceProtocol: HTTP
        LoadBalancerPort: 443
        Protocol: HTTPS
        PolicyNames:
          - Example-SSLNegotiation-Policy
        SSLCertificateId: !Ref 'ACMCertificate'
```

CT.ELASTICLOADBALANCING.PR.8 rule specification

# ###################################################################

782
## Rule Specification

### Rule Identifier:
elb_acm_certificate_required_check

### Description:
This control checks whether Classic Load Balancers use HTTPS/SSL certificates provided by AWS Certificate Manager.

### Reports on:
AWS::ElasticLoadBalancing::LoadBalancer

### Evaluates:
AWS CloudFormation, AWS CloudFormation hook

### Rule Parameters:
None

### Scenarios:

#### Scenario: 1
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document does not contain any Elastic Load Balancing load balancer resources
Then: SKIP

#### Scenario: 2
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elastic Load Balancing load balancer resource
And: There are no HTTPS or SSL 'Listeners' configured on the load balancer resource
Then: SKIP

#### Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elastic Load Balancing load balancer resource
And: There are one or more HTTPS or SSL 'Listeners' configured on the load balancer resource
And: 'SSLCertificateId' on load balancer HTTPS or SSL 'Listeners' is missing or not a valid ACM certificate ARN
Then: FAIL

#### Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elastic Load Balancing load balancer resource
And: There are one or more HTTPS or SSL 'Listeners' configured on the load balancer resource
And: 'SSLCertificateId' matches an ACM certificate ARN for all 'HTTPS' and 'SSL' 'Listeners'
Then: PASS

### Constants

```javascript
let ELASTIC_LOAD_BALANCER_TYPE = "AWS::ElasticLoadBalancing::LoadBalancer"
let ACM_CERTIFICATE_ARN_PATTERN = /arn:aws[a-z0-9-]*:acm:[a-z0-9-]+:d[12]:certificate/[^\w-](1,64)/
let SECURE_LISTENER_PROTOCOLS = ["HTTPS", "SSL"]
let INPUT_DOCUMENT = this
```

### Assignments

```javascript
let classic_load_balancers = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_TYPE ]
```
# Primary Rules

```
rule elb_acm_certificate_required_check when is_cfn_template(%INPUT_DOCUMENT)
  %classic_load_balancers not empty {
    check(%classic_load_balancers.Properties)
    <<
    [CT.ELASTICLOADBALANCING.PR.8]: Require any classic load balancer SSL/HTTPS
    listener to have a certificate provided by AWS Certificate Manager
    [FIX]: Configure Classic Load Balancers to use certificates provided by AWS
    Certificate Manager (ACM).
    >>
  }

rule elb_acm_certificate_required_check when is_cfn_hook(%INPUT_DOCUMENT,
  %ELASTIC_LOAD_BALANCER_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_TYPE.resourceProperties)
    <<
    [CT.ELASTICLOADBALANCING.PR.8]: Require any classic load balancer SSL/HTTPS
    listener to have a certificate provided by AWS Certificate Manager
    [FIX]: Configure Classic Load Balancers to use certificates provided by AWS
    Certificate Manager (ACM).
    >>
  }
```

# Parameterized Rules

```
rule check(classic_load_balancer) {
  %classic_load_balancer [
    filter_load_balancer_with_listeners(this)
  ] {
    Listeners [
      filter_secure_listeners(this)
    ] {
      # Scenarios 3 and 4
      SSLCertificateId exists
      SSLCertificateId == %ACM_CERTIFICATE_ARN_PATTERN or
      check_local_references(%INPUT_DOCUMENT, SSLCertificateId,
      "AWS::CertificateManager::Certificate")
    }
  }
}

rule filter_load_balancer_with_listeners(classic_load_balancer) {
  %classic_load_balancer {
    Listeners exists
    Listeners is_list
    Listeners not empty
  }
}

rule filter_secure_listeners(listener) {
  %listener {
    Protocol exists
    Protocol in %SECURE_LISTENER_PROTOCOLS
  }
}
```

# Utility Rules

```
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists  
```
CT.ELASTICLOADBALANCING.PR.8 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      - Ref: VPC
      CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      - Fn::Select:
        - 0
      - Fn::GetAZs: ''
SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      - Ref: VPC
      CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      - Fn::Select:
        - 1
      - Fn::GetAZs: ''
ACMCertificate:
  Type: "AWS::CertificateManager::Certificate"
  Properties:
    DomainName: example.com
    ValidationMethod: DNS
    DomainValidationOptions:
      - DomainName: www.example.com
        HostedZoneId: ZZZHHHHWWWAAA

ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
      - Ref: SubnetOne
      - Ref: SubnetTwo
    Policies:
      - PolicyName: Example-SSLNegotiation-Policy
        PolicyType: SSLNegotiationPolicyType
        Attributes:
          - Name: Reference-Security-Policy
            Value: ELBSecurityPolicy-TLS-1-2-2017-01
    Listeners:
      - InstancePort: '80'
        InstanceProtocol: HTTP
        LoadBalancerPort: '443'
        Protocol: HTTPS
        PolicyNames:
          - Example-SSLNegotiation-Policy
        SSLCertificateId:
          - Ref: ACMCertificate

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      - Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: ''
SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      - Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
        - 1
        - Fn::GetAZs: ''
ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
Properties:
  Scheme: internal
  Subnets:
  - Ref: SubnetOne
  - Ref: SubnetTwo
  Policies:
  - PolicyName: Example-SSLNegotiation-Policy
    PolicyType: SSLNegotiationPolicyType
    Attributes:
    - Name: Reference-Security-Policy
    - Value: ELBSecurityPolicy-TLS-1-2-2017-01
  Listeners:
  - InstancePort: '80'
    InstanceProtocol: HTTP
    LoadBalancerPort: '443'
    Protocol: HTTPS
    PolicyNames:
    - Example-SSLNegotiation-Policy
    SSLCertificateId: arn:aws:iam::123456789012:server-certificate/example-certificate

[CT.ELASTICLOADBALANCING.PR.9] Require that an AWS ELB Application or Classic Load Balancer listener is configured with HTTPS or TLS termination

This control checks whether your Elastic Load Balancing (ELB) Classic Load Balancer front-end listeners are configured with HTTPS or SSL protocols.

- **Control objective:** Encrypt data in transit
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElasticLoadBalancing::LoadBalancer
- **AWS CloudFormation guard rule:** [CT.ELASTICLOADBALANCING.PR.9 rule specification (p. 789)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICLOADBALANCING.PR.9 rule specification (p. 789)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICLOADBALANCING.PR.9 example templates (p. 791)]

Explanation

Before you start to use a load balancer, you must add one or more listeners. A listener is a process that uses the configured protocol and port to check for connection requests. Listeners can support HTTP and HTTPS/TLS protocols. You should always use an HTTPS or TLS listener, so that the load balancer does the work of encryption and decryption in transit.

Remediation for rule failure

Configure Classic Load Balancer front-end listeners with HTTPS or SSL protocols.

The examples that follow show how to implement this remediation.

**Classic Load Balancer - Example One**

Classic Load Balancer configured with an HTTPS Listener. The example is shown in JSON and in YAML.

**JSON example**
YAML example

LoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
      - !Ref 'Subnet'
  Listeners:
    - Protocol: HTTPS
      SSLCertificateId: !Ref 'ACMCertificate'
      InstancePort: 80
      LoadBalancerPort: 443

The examples that follow show how to implement this remediation.

Classic Load Balancer - Example Two

Classic Load Balancer configured with an SSL Listener. The example is shown in JSON and in YAML.

JSON example

```json
{
  "LoadBalancer": {
    "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
    "Properties": {
      "Scheme": "internal",
      "Subnets": [
        {
          "Ref": "Subnet"
        }
      ],
      "Listeners": [
        {
          "Protocol": "HTTPS",
          "SSLCertificateId": {
            "Ref": "ACMCertificate"
          },
          "InstancePort": 80,
          "LoadBalancerPort": 443
        }
      ]
    }
  }
}
```
"Listeners": [  
  {  
    "Protocol": "SSL",  
    "SSLCertificateId": {  
      "Ref": "ACMCertificate"  
    },  
    "InstancePort": 80,  
    "LoadBalancerPort": 443  
  }  
]  
}  
}  

YAML example

LoadBalancer:  
  Type: AWS::ElasticLoadBalancing::LoadBalancer  
  Properties:  
    Scheme: internal  
    Subnets:  
    - !Ref 'Subnet'  
    Listeners:  
      - Protocol: SSL  
        SSLCertificateId: !Ref 'ACMCertificate'  
        InstancePort: 80  
        LoadBalancerPort: 443  

CT.ELASTICLOADBALANCING.PR.9 rule specification

# ####################################################################  
##       Rule Specification        ##  
####################################################################  
# Rule Identifier:  
#   elb_tls_https_listeners_only_check  
# Description:  
#   Checks whether Classic Load Balancer front-end listeners are configured with HTTPS or SSL protocols.  
# Reports on:  
#   AWS::ElasticLoadBalancing::LoadBalancer  
# Evaluates:  
#   AWS CloudFormation, AWS CloudFormation hook  
# Rule Parameters:  
#   None  
# Scenarios:  
#   Scenario: 1  
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document  
#     And: The input document does not contain any Elastic Load Balancing LoadBalancer resources  
#     Then: SKIP  
#   Scenario: 2  

789
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancing LoadBalancer resource
# And: 'Listeners' has not been provided or is provided with a value of an empty list
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancing LoadBalancer resource
# And: 'Protocol' on LoadBalancer 'Listeners' is not set to 'HTTPS' or 'SSL'
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an ElasticLoadBalancing LoadBalancer resource
# And: 'Protocol' is set to 'HTTPS' or 'SSL' for all 'Listeners'
# Then: PASS

# Constants
#
let ELASTIC_LOAD_BALANCER_TYPE = "AWS::ElasticLoadBalancing::LoadBalancer"
let INPUT_DOCUMENT = this
#
# Assignments
#
let classic_load_balancers = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_TYPE ]
#
# Primary Rules
#
rule elb_tls_https_listeners_only_check when is_cfn_template(%INPUT_DOCUMENT)
%classic_load_balancers not empty {
  check(%classic_load_balancers.Properties)
  <<
  [CT.ELASTICLOADBALANCING.PR.9]: Require that an AWS ELB Application or Classic Load Balancer listener is configured with HTTPS or TLS termination
  [FIX]: Configure Classic Load Balancer front-end listeners with HTTPS or SSL protocols.
  >>
}

rule elb_tls_https_listeners_only_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_TYPE) {
  check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_TYPE.resourceProperties)
  <<
  [CT.ELASTICLOADBALANCING.PR.9]: Require that an AWS ELB Application or Classic Load Balancer listener is configured with HTTPS or TLS termination
  [FIX]: Configure Classic Load Balancer front-end listeners with HTTPS or SSL protocols.
  >>
}

# Parameterized Rules
#
rule check(classic_load_balancer) {
  %classic_load_balancer {
    # Scenario 2
    Listeners exists
    Listeners is_list
    Listeners not empty
# Scenarios 3 and 4

```python
Listeners[*] {
    Protocol exists
    Protocol in ["HTTPS", "SSL"]
}
}
```

# Utility Rules

```python
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

CT.ELASTICLOADBALANCING.PR.9 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```json
Resources:
VPC:
    Type: AWS::EC2::VPC
    Properties:
        CidrBlock: 10.0.0.0/16
        EnableDnsSupport: 'true'
        EnableDnsHostnames: 'true'
SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
        VpcId:
            Ref: VPC
        CidrBlock: 10.0.0.0/24
        AvailabilityZone:
            Fn::Select:
                - 0
                - Fn::GetAZs: ''
SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
        VpcId:
            Ref: VPC
        CidrBlock: 10.0.1.0/24
        AvailabilityZone:
            Fn::Select:
                - 1
                - Fn::GetAZs: ''
ClassicLoadBalancer:
    Type: AWS::ElasticLoadBalancing::LoadBalancer
    Properties:
        Scheme: internal
        Subnets:
            - Ref: SubnetOne
            - Ref: SubnetTwo
```
Listeners:
- Protocol: HTTPS
  SSLCertificateId: arn:aws:acm:us-east-1:123456789012:certificate/12345678-12ab-34cd-56ef-12345678
  InstancePort: 80
  LoadBalancerPort: 443

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: ''
SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
        - 1
        - Fn::GetAZs: ''
ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
      - Ref: SubnetOne
      - Ref: SubnetTwo
    Listeners:
      - Protocol: HTTP
        InstancePort: 80
        LoadBalancerPort: 80

[CT.ELASTICLOADBALANCING.PR.10] Require an ELB application or classic load balancer to have logging activated

This control checks whether Classic Load Balancers have logging enabled.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElasticLoadBalancing::LoadBalancer
• **AWS CloudFormation guard rule**: [CT.ELASTICLOADBALANCING.PR.10 rule specification (p. 794)]

**Details and examples**

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICLOADBALANCING.PR.10 rule specification (p. 794)]

• For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICLOADBALANCING.PR.10 example templates (p. 796)]

**Explanation**

Elastic Load Balancing provides access logs that capture detailed information about requests sent to your load balancer. Each log contains information such as the time the request was received, the client's IP address, latencies, request paths, and server responses. You can use these access logs to analyze traffic patterns and to troubleshoot issues.

**Remediation for rule failure**

Set an `AccessLoggingPolicy` and provide an `S3BucketName` with an Amazon S3 bucket configured to receive classic load balancer access logs.

The examples that follow show how to implement this remediation.

**Classic Load Balancer - Example**

Classic Load Balancer configured with an HTTPS listener and access logging. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ClassicLoadBalancer": {
        "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
        "Properties": {
            "Scheme": "internal",
            "Listeners": [
                {
                    "Protocol": "HTTPS",
                    "InstancePort": 80,
                    "LoadBalancerPort": 443
                }
            ],
            "Subnets": [
                {
                    "Ref": "Subnet"
                }
            ],
            "AccessLoggingPolicy": {
                "Enabled": true,
                "S3BucketName": {
                    "Ref": "LoggingBucket"
                }
            }
        }
    }
}
```

**YAML example**

```yaml

```

793
ClassicLoadBalancer:
Type: AWS::ElasticLoadBalancing::LoadBalancer
Properties:
  Scheme: internal
  Listeners:
    - Protocol: HTTPS
      InstancePort: 80
      LoadBalancerPort: 443
  Subnets:
    - !Ref 'Subnet'
  AccessLoggingPolicy:
    Enabled: true
    S3BucketName: !Ref 'LoggingBucket'

CT.ELASTICLOADBALANCING.PR.10 rule specification

# #####################################################################
##       Rule Specification        ##
#####################################################################
# # Rule Identifier: #
# elb_logging_enabled_check #
# # Description: #
# This control checks whether Classic Load Balancers have logging enabled. #
# # Reports on: #
# AWS::ElasticLoadBalancing::LoadBalancer #
# # Evaluates: #
# AWS CloudFormation, AWS CloudFormation hook #
# # Rule Parameters: #
# None #
# # Scenarios: #
# Scenario: 1 #
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook #
# document #
# And: The input document does not contain any Elastic Load Balancing load balancer #
# resources #
# Then: SKIP #
# Scenario: 2 #
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook #
# document #
# And: The input document contains an Elastic Load Balancing load balancer resource #
# And: 'AccessLoggingPolicy' has not been provided #
# Then: FAIL #
# Scenario: 3 #
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook #
# document #
# And: The input document contains an Elastic Load Balancing load balancer resource #
# And: 'AccessLoggingPolicy' has been provided #
# And: 'Enabled' in 'AccessLoggingPolicy' is missing or has been set to bool(false) #
# or 'S3BucketName' is missing #
# Then: FAIL #
# Scenario: 4 #
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook #
# document
And: The input document contains an Elastic Load Balancing load balancer resource
And: 'AccessLoggingPolicy' has been provided
And: 'Enabled' has been provided in 'AccessLoggingPolicy' and has been set to
    bool(true)
And: 'S3BucketName' has been provided in 'AccessLoggingPolicy' as a non-empty
    string value or
    valid local reference
Then: PASS

Constants

let ELASTIC_LOAD_BALANCER_TYPE = "AWS::ElasticLoadBalancing::LoadBalancer"
let INPUT_DOCUMENT = this

Assignments

let classic_load_balancers = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_TYPE ]

Primary Rules

rule elb_logging_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
  %classic_load_balancers not empty {
    check(%classic_load_balancers.Properties)
    <<
      [CT.ELASTICLOADBALANCING.PR.10]: Require an ELB application or classic load
      balancer to have logging activated
      [FIX]: Set an 'AccessLoggingPolicy' and provide an 'S3BucketName' with an Amazon S3
      bucket configured to receive classic load balancer access logs.
    >>
  }

rule elb_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_TYPE) {
  check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_TYPE.resourceProperties)
  <<
    [CT.ELASTICLOADBALANCING.PR.10]: Require an ELB application or classic load
    balancer to have logging activated
    [FIX]: Set an 'AccessLoggingPolicy' and provide an 'S3BucketName' with an Amazon S3
    bucket configured to receive classic load balancer access logs.
  >>
}

Parameterized Rules

rule check(classic_load_balancer) {
  %classic_load_balancer {
    # Scenario 2
    AccessLoggingPolicy exists
    AccessLoggingPolicy is_struct
    AccessLoggingPolicy {
      # Scenario 3 and 4
      Enabled exists
      Enabled == true
      S3BucketName exists
      check_is_string_and_not_empty(S3BucketName) or
      check_local_references(%INPUT_DOCUMENT, S3BucketName, "AWS::S3::Bucket")
    }
  }
}
CT.ELASTICLOADBALANCING.PR.10 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Mappings:
RegionToELBAccountId:
us-east-1:
  AccountId: '127311923021'
us-west-1:
  AccountId: '027434742980'
us-west-2:
  AccountId: '797873946194'
ca-central-1:
  AccountId: '985666609251'
eu-west-1:
  AccountId: '156460612806'
ap-northeast-1:
  AccountId: '582318560864'
ap-northeast-2:
    AccountId: '600734575887'
ap-southeast-1:
    AccountId: '114774131450'
ap-southeast-2:
    AccountId: '783225319266'
ap-south-1:
    AccountId: '718504428378'
us-east-2:
    AccountId: '033677994240'
sa-east-1:
    AccountId: '507241528517'
eu-central-1:
    AccountId: '054676820928'
ar-south-1:
    AccountId: '098369216593'
ap-east-1:
    AccountId: '754344486484'
ap-southeast-3:
    AccountId: '589379963580'
ap-northeast-3:
    AccountId: '383597477331'
eu-west-2:
    AccountId: '652711504416'
eu-south-1:
    AccountId: '635631232127'
eu-west-3:
    AccountId: '009996457667'
eu-north-1:
    AccountId: '897822967062'
me-south-1:
    AccountId: '376674570225'
us-gov-west-1:
    AccountId: '048591011584'
us-gov-east-1:
    AccountId: '190560391635'
RegionToARNPrefix:
    us-east-1:
        ARNPrefix: 'arn:aws:'
    us-west-1:
        ARNPrefix: 'arn:aws:'
    us-west-2:
        ARNPrefix: 'arn:aws:'
    ca-central-1:
        ARNPrefix: 'arn:aws:'
    eu-west-1:
        ARNPrefix: 'arn:aws:'
    ap-northeast-1:
        ARNPrefix: 'arn:aws:'
    ap-northeast-2:
        ARNPrefix: 'arn:aws:'
    ap-southeast-1:
        ARNPrefix: 'arn:aws:'
    ap-southeast-2:
        ARNPrefix: 'arn:aws:'
    ap-south-1:
        ARNPrefix: 'arn:aws:'
    us-east-2:
        ARNPrefix: 'arn:aws:'
    sa-east-1:
        ARNPrefix: 'arn:aws:'
    eu-central-1:
        ARNPrefix: 'arn:aws:'
    af-south-1:
        ARNPrefix: 'arn:aws:'
    ap-east-1:
ARNPrefix: 'arn:aws:'
ap-southeast-3:
  ARNPrefix: 'arn:aws:'
ap-northeast-3:
  ARNPrefix: 'arn:aws:'
eu-west-2:
  ARNPrefix: 'arn:aws:'
eu-south-1:
  ARNPrefix: 'arn:aws:'
eu-west-3:
  ARNPrefix: 'arn:aws:'
eu-north-1:
  ARNPrefix: 'arn:aws:'
me-south-1:
  ARNPrefix: 'arn:aws:'
us-gov-west-1:
  ARNPrefix: 'arn:aws-us-gov:'
us-gov-east-1:
  ARNPrefix: 'arn:aws-us-gov:'

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
      - 0
      - Fn::GetAZs: ''
SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
      - 1
      - Fn::GetAZs: ''
LoggingBucket:
  Type: AWS::S3::Bucket
LoggingBucketPolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    Bucket:
      Ref: LoggingBucket
    PolicyDocument:
      Version: 2012-10-17
      Statement:
      - Action:
        - 's3:PutObject'
        Effect: Allow
        Resource:
          Fn::Join:
          - ''
          - Fn::FindInMap: [RegionToARNPrefix, !Ref 'AWS::Region', ARNPrefix]
          - 's3:::'
          - Ref: LoggingBucket
          - /AWSLogs/
Proactive controls

- Ref: AWS::AccountId
- /*
  Principal:
  AWS:
  Fn::FindInMap: [RegionToELBAccountId, !Ref 'AWS::Region', AccountId]

ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Listeners:
      - Protocol: HTTP
        InstancePort: 80
        LoadBalancerPort: 80
    Subnets:
      - Ref: SubnetOne
      - Ref: SubnetTwo
    AccessLoggingPolicy:
      Enabled: true
      S3BucketName:
        Ref: LoggingBucket

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
  SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/24
      AvailabilityZone: 
        Fn::Select:
          - 0
          - Fn::GetAZs: ''
  SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.1.0/24
      AvailabilityZone: 
        Fn::Select:
          - 1
          - Fn::GetAZs: ''

ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Listeners:
      - Protocol: HTTP
        InstancePort: 80
        LoadBalancerPort: 80
    Subnets:
      - Ref: SubnetOne
      - Ref: SubnetTwo
[CT.ELASTICLOADBALANCING.PR.11] Require any ELB classic load balancer to have connection draining activated

This control checks whether Elastic Load Balancing (ELB) Classic Load Balancers have connection draining configured.

- **Control objective:** Improve resiliency
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElasticLoadBalancing::LoadBalancer
- **AWS CloudFormation guard rule:** CT.ELASTICLOADBALANCING.PR.11 rule specification (p. 801)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.ELASTICLOADBALANCING.PR.11 rule specification (p. 801)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.ELASTICLOADBALANCING.PR.11 example templates (p. 803)

Explanation

Activating connection draining on Classic Load Balancers ensures that the load balancer stops sending requests to instances that are de-registering or unhealthy. It keeps the existing connections open. This configuration is particularly useful for instances in Auto Scaling groups, to ensure that connections aren't severed abruptly.

**Remediation for rule failure**


The examples that follow show how to implement this remediation.

**Classic Load Balancer - Example**

Classic Load Balancer configured with connection draining active. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ClassicLoadBalancer": {
        "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
        "Properties": {
            "Scheme": "internal",
            "Listeners": [
                {
                    "InstancePort": "80",
                    "InstanceProtocol": "HTTP",
                    "LoadBalancerPort": "443",
                    "Protocol": "HTTPS",
                    "PolicyNames": [
                        "Example-SSLNegotiation-Policy"
                    ],
                    "SSLCertificateId": {
                        "Ref": "ACMCertificate"
                    }
                }
            ]
        }
    }
}
```
YAML example

ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Listeners:
      - InstancePort: '80'
        InstanceProtocol: HTTP
        LoadBalancerPort: '443'
        Protocol: HTTPS
        PolicyNames:
          - Example-SSLNegotiation-Policy
        SSLCertificateId: !Ref 'ACMCertificate'
    Policies:
      - PolicyName: Example-SSLNegotiation-Policy
        PolicyType: SSLNegotiationPolicyType
        Attributes:
          - Name: Reference-Security-Policy
            Value: ELBSecurityPolicy-TLS-1-2-2017-01
    Subnets:
      - !Ref 'SubnetOne'
      - !Ref 'SubnetTwo'
    ConnectionDrainingPolicy:
      Enabled: true
Proactive controls

## Rule Specification

### Rule Identifier:
- `elb_connection_draining_enabled_check`

### Description:
- This control checks whether Elastic Load Balancing (ELB) Classic Load Balancers have connection draining configured.

### Reports on:
- `AWS::ElasticLoadBalancing::LoadBalancer`

### Evaluates:
- AWS CloudFormation, AWS CloudFormation hook

### Rule Parameters:
- None

### Scenarios:

#### Scenario: 1
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document does not contain any Elastic Load Balancing load balancer resources
- Then: SKIP

#### Scenario: 2
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document contains an Elastic Load Balancing load balancer resource
- And: 'ConnectionDrainingPolicy' has not been specified
- Then: FAIL

#### Scenario: 3
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document contains an Elastic Load Balancing load balancer resource
- And: 'ConnectionDrainingPolicy' has been specified
- And: 'Enabled' in 'ConnectionDrainingPolicy' is missing or has been set to bool(false)
- Then: FAIL

#### Scenario: 4
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- And: The input document contains an Elastic Load Balancing load balancer resource
- And: 'ConnectionDrainingPolicy' has been specified
- And: 'Enabled' in 'ConnectionDrainingPolicy' has been set to bool(true)
- Then: PASS

### Constants

let ELASTIC_LOAD_BALANCER_TYPE = "AWS::ElasticLoadBalancing::LoadBalancer"
let INPUT_DOCUMENT = this

### Assignments

let classic_load_balancers = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_TYPE ]

### Primary Rules

rule elb_connection_draining_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%classic_load_balancers not empty {
    check(%classic_load_balancers.Properties)
}

```
<<

[CT.ELASTICLOADBALANCING.PR.11]: Require any ELB classic load balancer to have connection draining activated

[FIX]: Configure a 'ConnectionDrainingPolicy' on Elastic Load Balancing Classic Load Balancers.

>>

rule elb_connection_draining_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_TYPE.resourceProperties)
    <<
    [CT.ELASTICLOADBALANCING.PR.11]: Require any ELB classic load balancer to have connection draining activated
    [FIX]: Configure a 'ConnectionDrainingPolicy' on Elastic Load Balancing Classic Load Balancers.
    >>
}

# Parameterized Rules

# rule check(classic_load_balancer) {
#     %classic_load_balancer {
#         # Scenario 2
#         ConnectionDrainingPolicy exists
#         ConnectionDrainingPolicy is_struct
#         ConnectionDrainingPolicy {
#             # Scenario 3 and 4
#             Enabled exists
#             Enabled == true
#         }
#     }
# }

# Utility Rules

# rule is_cfn_template(doc) {
#     %doc {
#         AWSTemplateFormatVersion exists or Resources exists
#     }
# }

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.ELASTICLOADBALANCING.PR.11 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
    Type: AWS::EC2::VPC
    Properties:
        CidrBlock: 10.0.0.0/16
        EnableDnsSupport: 'true'
EnableDnsHostnames: 'true'

SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
        - 0
      - Fn::GetAZs: ''

SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
        - 1
      - Fn::GetAZs: ''

ACMCertificate:
  Type: "AWS::CertificateManager::Certificate"
  Properties:
    DomainName: example.com
    ValidationMethod: DNS
    DomainValidationOptions:
      - DomainName: www.example.com
      - HostedZoneId: ZZZHHHHAAAAA

ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Listeners:
      - InstancePort: '80'
        InstanceProtocol: HTTP
        LoadBalancerPort: '443'
        Protocol: HTTPS
        PolicyNames:
          - Example-SSLNegotiation-Policy
        SSLCertificateId:
          Ref: ACMCertificate
        Policies:
          - PolicyName: Example-SSLNegotiation-Policy
            PolicyType: SSLNegotiationPolicyType
            Attributes:
              - Name: Reference-Security-Policy
                Value: ELBSecurityPolicy-TLS-1-2-2017-01
        Subnets:
          - Ref: SubnetOne
          - Ref: SubnetTwo
        ConnectionDrainingPolicy:
          Enabled: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
EnableDnsHostnames: 'true'

SubnetOne:
Type: AWS::EC2::Subnet
Properties:
  VpcId: Ref: VPC
  CidrBlock: 10.0.0.0/24
  AvailabilityZone: 
    Fn::Select:
      - 0
      - Fn::GetAZs: ''

SubnetTwo:
Type: AWS::EC2::Subnet
Properties:
  VpcId: Ref: VPC
  CidrBlock: 10.0.1.0/24
  AvailabilityZone: 
    Fn::Select:
      - 1
      - Fn::GetAZs: ''

ClassicLoadBalancer:
Type: AWS::ElasticLoadBalancing::LoadBalancer
Properties:
  Scheme: internal
  Listeners:
    - Protocol: HTTP
      InstancePort: 80
      LoadBalancerPort: 80
  Subnets:
    - Ref: SubnetOne
    - Ref: SubnetTwo

[CT.ELASTICLOADBALANCING.PR.12] Require any ELB classic load balancer SSL/HTTPS listener to have a predefined security policy with a strong configuration

This control checks whether Elastic Load Balancing (ELB) Classic Load Balancer HTTPS/SSL listeners use the predefined security policy ELBSecurityPolicy-TLS-1-2-2017-01.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElasticLoadBalancing::LoadBalancer
- **AWS CloudFormation guard rule:** [CT.ELASTICLOADBALANCING.PR.12 rule specification](p. 807)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICLOADBALANCING.PR.12 rule specification](p. 807)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICLOADBALANCING.PR.12 example templates](p. 810)

Explanation

A security policy is a combination of SSL protocols, ciphers, and the server order preference option. Predefined policies control the ciphers, protocols, and preference orders that provide support during SSL negotiations between a client and load balancer.
Using ELBSecurityPolicy-TLS-1-2-2017-01 can help you to meet compliance and security standards that require you to turn off specific versions of SSL and TLS.

**Usage considerations**

- This control applies only to Elastic Load Balancing Classic Load Balancers configured with HTTPS or SSL listeners.

**Remediation for rule failure**

Configure Classic Load Balancer HTTPS/SSL listeners to use the predefined security policy called ELBSecurityPolicy-TLS-1-2-2017-01.

The examples that follow show how to implement this remediation.

**Classic Load Balancer - Example**

Classic Load Balancer configured with an HTTPS listener and SSL negotiation policy that references the ELBSecurityPolicy-TLS-1-2-2017-01 predefined security policy for classic load balancers. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ClassicLoadBalancer": {
        "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
        "Properties": {
            "Scheme": "internal",
            "Subnets": [
                {
                    "Ref": "SubnetOne"
                },
                {
                    "Ref": "SubnetTwo"
                }
            ],
            "Policies": [
                {
                    "PolicyName": "Example-SSLNegotiation-Policy",
                    "PolicyType": "SSLNegotiationPolicyType",
                    "Attributes": [
                        {
                            "Name": "Reference-Security-Policy",
                            "Value": "ELBSecurityPolicy-TLS-1-2-2017-01"
                        }
                    ]
                }
            ],
            "Listeners": [
                {
                    "InstancePort": 80,
                    "InstanceProtocol": "HTTP",
                    "LoadBalancerPort": 443,
                    "Protocol": "HTTPS",
                    "SSLCertificateId": {
                        "Ref": "ACMCertificate"
                    },
                    "PolicyNames": [
                        "Example-SSLNegotiation-Policy"
                    ]
                }
            ]
    }
}
```
YAML example

ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
      - !Ref 'SubnetOne'
      - !Ref 'SubnetTwo'
    Policies:
      - PolicyName: Example-SSLNegotiation-Policy
        PolicyType: SSLNegotiationPolicyType
        Attributes:
          - Name: Reference-Security-Policy
            Value: ELBSecurityPolicy-TLS-1-2-2017-01
    Listeners:
      - InstancePort: 80
        InstanceProtocol: HTTP
        LoadBalancerPort: 443
        Protocol: HTTPS
        SSLCertificateId: !Ref 'ACMCertificate'
        PolicyNames:
          - Example-SSLNegotiation-Policy

CT.ELASTICLOADBALANCING.PR.12 rule specification

# ###################################################################
## Rule Specification ##
# ###################################################################
# # Rule Identifier: # elb_predefined_security_policy_ssl_check # # # Description: # This control checks whether Elastic Load Balancing (ELB) Classic Load Balancer HTTPS/SSL listeners use the predefined security policy 'ELBSecurityPolicy-TLS-1-2-2017-01'. # # Reports on: # AWS::ElasticLoadBalancing::LoadBalancer # # Evaluates: # AWS CloudFormation, AWS CloudFormation hook # # Rule Parameters: # None # # Scenarios: # Scenario: 1 #   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document #   And: The input document does not contain any Elastic Load Balancing LoadBalancer resources #   Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
# document
#     And: The input document contains an Elastic Load Balancing LoadBalancer resource
#     And: There are no HTTPS or SSL 'Listeners' configured on the Elastic Load Balancing
# LoadBalancer resource
#     Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
# document
#     And: The input document contains an Elastic Load Balancing LoadBalancer resource
#     And: 'Policies' does not contain a policy with 'PolicyType' equal to
# 'SSLNegotiationPolicyType'
#     Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
# document
#     And: The input document contains an Elastic Load Balancing LoadBalancer resource
#     And: 'Policies' contains a policy with 'PolicyType' equal to
# 'SSLNegotiationPolicyType'
#     And: 'Policies' is missing a 'Reference-Security-Policy' with a value of
# 'ELBSecurityPolicy-TLS-1-2-2017-01'
#     Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
# document
#     And: The input document contains an Elastic Load Balancing LoadBalancer resource
#     And: 'Policies' contains a policy with 'PolicyType' equal to
# 'SSLNegotiationPolicyType'
#     And: 'Policies' contains a 'Reference-Security-Policy' with a value of
# 'ELBSecurityPolicy-TLS-1-2-2017-01'
#     And: A 'HTTPS' or 'SSL' Listener on the LoadBalancer resource does not reference
# the secure policy
#     Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
# document
#     And: The input document contains an Elastic Load Balancing LoadBalancer resource
#     And: 'Policies' contains a policy with 'PolicyType' equal to
# 'SSLNegotiationPolicyType'
#     And: 'Policies' contains a 'Reference-Security-Policy' with a value of
# 'ELBSecurityPolicy-TLS-1-2-2017-01'
#     And: All 'HTTPS' and 'SSL' Listeners on the LoadBalancer resource reference the
# secure policy
#     Then: PASS

# Constants
let ELASTIC_LOAD_BALANCER_TYPE = "AWS::ElasticLoadBalancing::LoadBalancer"
let VALID_REFERENCE_SECURITY_POLICIES = [ "ELBSecurityPolicy-TLS-1-2-2017-01" ]
let INPUT_DOCUMENT = this

# Assignments
let classic_load_balancers = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_TYPE ]

# Primary Rules
# rule elb_predefined_security_policy_ssl_check when is_cfn_template(%INPUT_DOCUMENT)
check(%classic_load_balancers.Properties)
[CT.ELASTICLOADBALANCING.PR.12]: Require any ELB classic load balancer SSL/HTTPS listener to have a predefined security policy with a strong configuration

[Fix]: Configure classic load balancer HTTPS/SSL listeners to use the predefined security policy called ELBSecurityPolicy-TLS-1-2-2017-01.

```python
rule elb_predefined_security_policy_ssl_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_TYPE.resourceProperties)
    <<<
    [CT.ELASTICLOADBALANCING.PR.12]: Require any ELB classic load balancer SSL/HTTPS listener to have a predefined security policy with a strong configuration
    [Fix]: Configure classic load balancer HTTPS/SSL listeners to use the predefined security policy called ELBSecurityPolicy-TLS-1-2-2017-01.
    >>>
}

# Parameterized Rules

rule check(classic_load_balancer) {
    %classic_load_balancer {
        let elb = this

        # Scenario 2
        Listeners[Protocol in ["HTTPS", "SSL"]] {
            %elb.Policies exists
            %elb.Policies is_list
            %elb.Policies not empty

            let secure_policies = %elb.Policies[
                PolicyType == "SSLNegotiationPolicyType"
                some Attributes[*] {
                    Name == "Reference-Security-Policy"
                    Value in %VALID_REFERENCE_SECURITY_POLICIES
                }
            ].PolicyName

            # Scenarios 3 and 4
            %secure_policies not empty

            # Scenarios 5 and 6
            PolicyNames exists
            PolicyNames is_list
            PolicyNames not empty
            some PolicyNames.* in %secure_policies
        }
    }
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.ELASTICLOADBALANCING.PR.12 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'

SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
      - 0
      - Fn::GetAZs: ''

SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
      - 1
      - Fn::GetAZs: ''

ACMCertificate:
  Type: "AWS::CertificateManager::Certificate"
  Properties:
    DomainName: example.com
    ValidationMethod: DNS
    DomainValidationOptions:
      - DomainName: www.example.com
        HostedZoneId: ZZZHHHHWWWWAAA

ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
      - Ref: SubnetOne
      - Ref: SubnetTwo
    Policies:
      - PolicyName: Example-SSLNegotiation-Policy
        PolicyType: SSLNegotiationPolicyType
        Attributes:
          - Name: Reference-Security-Policy
            Value: ELBSecurityPolicy-TLS-1-2-2017-01
    Listeners:
      - InstancePort: 80
        InstanceProtocol: HTTP
        LoadBalancerPort: 443
        Protocol: HTTPS
        SSLSertificateId:
          Ref: ACMCertificate
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone: 
      Fn::Select:
        - 0
        - Fn::GetAZs: ''
SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone: 
      Fn::Select:
        - 1
        - Fn::GetAZs: ''
ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Subnets:
      - Ref: SubnetOne
      - Ref: SubnetTwo
    Policies:
    - PolicyName: Example-SSLNegotiation-Policy
      PolicyType: SSLNegotiationPolicyType
      Attributes:
        - Name: Reference-Security-Policy
        Value: ELBSecurityPolicy-2016-08
    Listeners:
    - InstancePort: 80
      InstanceProtocol: HTTP
      LoadBalancerPort: 443
      Protocol: HTTPS
      SslCertificateId: arn:aws:iam::123456789012:server-certificate/example-certificate
      PolicyNames:
        - Example-SSLNegotiation-Policy

[CT.ELASTICLOADBALANCING.PR.13] Require any ELB classic load balancer to have cross-zone load balancing activated

This control checks whether cross-zone load balancing is configured for your Classic Load Balancer.
• **Control objective:** Improve availability

• **Implementation:** AWS CloudFormation Guard Rule

• **Control behavior:** Proactive

• **Resource types:** AWS::ElasticLoadBalancing::LoadBalancer

• **AWS CloudFormation guard rule:** [CT.ELASTICLOADBALANCING.PR.13 rule specification](#)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICLOADBALANCING.PR.13 rule specification](#)

- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICLOADBALANCING.PR.13 example templates](#)

**Explanation**

A load balancer node distributes traffic across the registered targets in its Availability Zone. When cross-zone load balancing is turned off, each load balancer node distributes traffic only across the registered targets in its own Availability Zone. If the number of registered targets is not same across the Availability Zones, traffic is not distributed evenly, so the instances in one zone may become over-utilized, when compared to the instances in another zone. With cross-zone load balancing activated, each load balancer node for your classic load balancer distributes requests evenly across the registered instances in all enabled Availability Zones.

**Remediation for rule failure**

Set CrossZone to true on Classic Load Balancers.

The examples that follow show how to implement this remediation.

**Classic Load Balancer - Example**

Classic Load Balancer configured with cross-zone load balancing active. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ClassicLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
    "Properties": {
      "Scheme": "internal",
      "Listeners": [
        {
          "InstancePort": "80",
          "InstanceProtocol": "HTTP",
          "LoadBalancerPort": "443",
          "Protocol": "HTTPS",
          "PolicyNames": [
            "Sample-SSLNegotiation-Policy"
          ],
          "SSLCertificateId": {
            "Ref": "ACMCertificate"
          }
        }
      ],
      "Policies": [
        {
```
YAML example

ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Listeners:
      - InstancePort: '80'
        InstanceProtocol: HTTP
        LoadBalancerPort: '443'
        Protocol: HTTPS
        PolicyNames:
          - Sample-SSLNegotiation-Policy
    SSLCertificateId: !Ref 'ACMCertificate'
  Policies:
    - PolicyName: Sample-SSLNegotiation-Policy
      PolicyType: SSLNegotiationPolicyType
      Attributes:
        - Name: Reference-Security-Policy
          Value: ELBSecurityPolicy-TLS-1-2-2017-01
    Subnets:
      - !Ref 'SubnetOne'
      - !Ref 'SubnetTwo'
  CrossZone: true

CT.ELASTICLOADBALANCING.PR.13 rule specification

# ###################################
##       Rule Specification        ##
####################################
#
# Rule Identifier:
#   elb_cross_zone_load_balancing_enabled_check
# Description:
This control checks whether cross-zone load balancing is configured for your Classic Load Balancer.

Reports on:
- AWS::ElasticLoadBalancing::LoadBalancer

Evaluates:
- AWS CloudFormation, AWS CloudFormation hook

Rule Parameters:
- None

Scenarios:

Scenario: 1
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document does not contain any Elastic Load Balancing LoadBalancer resources
Then: SKIP

Scenario: 2
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elastic Load Balancing LoadBalancer resource
And: 'CrossZone' has not been specified
Then: FAIL

Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elastic Load Balancing LoadBalancer resource
And: 'CrossZone' has been specified and set to bool(false)
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elastic Load Balancing LoadBalancer resource
And: 'CrossZone' has been specified and set to bool(true)
Then: PASS

Constants

let ELASTIC_LOAD_BALANCER_TYPE = "AWS::ElasticLoadBalancing::LoadBalancer"
let INPUT_DOCUMENT = this

Assignments

let classic_load_balancers = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_TYPE ]

Primary Rules

rule elb_cross_zone_load_balancing_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%classic_load_balancers not empty {
    check(%classic_load_balancers.Properties)
    <<
    [CT.ELASTICLOADBALANCING.PR.13]: Require any ELB classic load balancer to have cross-zone load balancing activated
    [FIX]: Set 'CrossZone' to 'true' on Classic Load Balancers.
    >>
}

rule elb_cross_zone_load_balancing_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_TYPE.resourceProperties)
    <<
## CT.ELASTICLOADBALANCING.PR.13 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
  SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/24
      AvailabilityZone: Fn::Select:
        - 0
        - Fn::GetAZs: '
  SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
```

---

[CT.ELASTICLOADBALANCING.PR.13]: Require any ELB classic load balancer to have cross-zone load balancing activated

[FIX]: Set 'CrossZone' to 'true' on Classic Load Balancers.

---

## Parameterized Rules

```yaml
rule check(classic_load_balancer) {
  %classic_load_balancer {
    # Scenario 2
    CrossZone exists
      # Scenario 3 and 4
      CrossZone == true
  }
}
```

## Utility Rules

```yaml
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or Resources exists
  }
}
```

```yaml
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```
CidrBlock: 10.0.1.0/24
AvailabilityZone:
  - Fn::Select:
  - 1
  - Fn::GetAZs: ''
ACMCertificate:
  Type: "AWS::CertificateManager::Certificate"
  Properties:
    DomainName: example.com
    ValidationMethod: DNS
    DomainValidationOptions:
      - DomainName: www.example.com
        HostedZoneId: ZZZHHHHWWWWAAA
ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Listeners:
      - InstancePort: '80'
        InstanceProtocol: HTTP
        LoadBalancerPort: '443'
        Protocol: HTTPS
        PolicyNames:
          - Example-SSLNegotiation-Policy
          SSLCertificateId:
            Ref: ACMCertificate
        Policies:
          - PolicyName: Example-SSLNegotiation-Policy
            PolicyType: SSLNegotiationPolicyType
            Attributes:
              - Name: Reference-Security-Policy
                Value: ELBSecurityPolicy-TLS-1-2-2017-01
      Subnets:
        - Ref: SubnetOne
        - Ref: SubnetTwo
    CrossZone: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'
SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      - Fn::Select:
        - 0
      - Fn::GetAZs: ''
SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
[CT.ELASTICLOADBALANCING.PR.14] Require a Network Load Balancer to have cross-zone load balancing activated

This control checks whether a Network Load Balancer (NLB) is configured with cross-zone load balancing.

- **Control objective:** Improve resiliency, Improve availability
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::ElasticLoadBalancingV2::LoadBalancer
- **AWS CloudFormation guard rule:** [CT.ELASTICLOADBALANCING.PR.14 rule specification](p. 818)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICLOADBALANCING.PR.14 rule specification](p. 818)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICLOADBALANCING.PR.14 example templates](p. 821)

**Explanation**

The nodes for your load balancer distribute requests from clients to registered targets. When cross-zone load balancing is enabled, each load balancer node distributes traffic across the registered targets in all enabled Availability Zones. When cross-zone load balancing is not enabled, each load balancer node distributes traffic only across the registered targets in its own Availability Zone.

**Usage considerations**

- This control applies only to a Network Load Balancer (Type of network).
- With a Network Load Balancer, cross-zone load balancing is off by default at the load-balancer level. You can turn it on at any time. For target groups, the default is to use the load balancer setting, but you can override the default by turning cross-zone load balancing on or off explicitly, at the target group level. To ensure that cross-zone load balancing is configured on target groups, use this control in conjunction with CT.ELASTICLOADBALANCING.PR.15.

**Remediation for rule failure**

Set the load balancer attribute `load_balancing.cross_zone.enabled` to `true`.

---

```yaml
AvailabilityZone:
  Fn::Select:
  - 1
  - Fn::GetAZs: ''

ClassicLoadBalancer:
  Type: AWS::ElasticLoadBalancing::LoadBalancer
  Properties:
    Scheme: internal
    Listeners:
    - Protocol: HTTP
      InstancePort: 80
      LoadBalancerPort: 80
    Subnets:
    - Ref: SubnetOne
    - Ref: SubnetTwo
    CrossZone: false
```
The examples that follow show how to implement this remediation.

**Network Load Balancer - Example**

Network Load Balancer configured with cross-zone load balancing enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "NetworkLoadBalancer": {
      "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
      "Properties": {
         "Scheme": "internal",
         "Type": "network",
         "Subnets": [
            {
               "Ref": "SubnetOne"
            },
            {
               "Ref": "SubnetTwo"
            }
         ],
         "IpAddressType": "ipv4",
         "LoadBalancerAttributes": [
            {
               "Key": "load_balancing.cross_zone.enabled",
               "Value": true
            }
         ]
      }
   }
}
```

**YAML example**

```
NetworkLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internal
    Type: network
    Subnets:
      - !Ref 'SubnetOne'
      - !Ref 'SubnetTwo'
    IpAddressType: ipv4
    LoadBalancerAttributes:
      - Key: load_balancing.cross_zone.enabled
        Value: true
```

**CT.ELASTICLOADBALANCING.PR.14 rule specification**

```
# ##################################################################
## Rule Specification     ##
# ##################################################################
# # Rule Identifier:
```
# nlb_cross_zone_load_balancing_enabled_check

## Description:
This control checks whether a Network Load Balancer (NLB) is configured with cross-zone load balancing.

## Reports on:
- AWS::ElasticLoadBalancingV2::LoadBalancer

## Evaluates:
- AWS CloudFormation, AWS CloudFormation hook

## Rule Parameters:
None

## Scenarios:

#### Scenario: 1
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document does not contain any ELBv2 Load Balancer resources
  - Then: SKIP

#### Scenario: 2
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document contains an ELBv2 Load Balancer resource
  - And: 'Type' has been provided and is set to a value other than 'network'
  - Then: SKIP

#### Scenario: 3
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document contains an ELBv2 Load Balancer resource
  - And: 'Type' has been provided and set to 'network'
  - And: 'LoadBalancerAttributes' has not been provided or has been provided as an empty list
  - Then: FAIL

#### Scenario: 4
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document contains an ELBv2 Load Balancer resource
  - And: 'Type' has been provided and set to 'network'
  - And: 'LoadBalancerAttributes' has been provided as a non-empty list
  - Then: FAIL

#### Scenario: 5
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document contains an ELBv2 Load Balancer resource
  - And: 'Type' has been provided and set to 'network'
  - And: 'LoadBalancerAttributes' has been provided as a non-empty list
  - And: 'LoadBalancerAttributes' contains an entry with a 'Key' equal to 'load_balancing.cross_zone.enabled'
  - Then: FAIL

#### Scenario: 6
- Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
  - And: The input document contains an ELBv2 Load Balancer resource
  - And: 'Type' has been provided and set to 'network'
  - And: 'LoadBalancerAttributes' has been provided as a non-empty list
  - And: 'LoadBalancerAttributes' contains an entry with a 'Key' equal to 'load_balancing.cross_zone.enabled'
  - Then: PASS

#
# Constants
#
let ELASTIC_LOAD_BALANCER_V2_TYPE = "AWS::ElasticLoadBalancingV2::LoadBalancer"
let INPUT_DOCUMENT = this
#
# Assignments
#
let elastic_load_balancers = Resources.*[ Type == %ELASTIC_LOAD_BALANCER_V2_TYPE ]
#
# Primary Rules
#
rule nlb_cross_zone_load_balancing_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
  %elastic_load_balancers not empty {
    check(%elastic_load_balancers.Properties)
    <<
      [CT.ELASTICLOADBALANCING.PR.14]: Require a Network Load Balancer to have cross-zone
      load balancing activated
      [FIX]: Set the load balancer attribute 'load_balancing.cross_zone.enabled' to 'true'.
    >>
}

rule nlb_cross_zone_load_balancing_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTIC_LOAD_BALANCER_V2_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTIC_LOAD_BALANCER_V2_TYPE.resourceProperties)
    <<
      [CT.ELASTICLOADBALANCING.PR.14]: Require a Network Load Balancer to have cross-zone
      load balancing activated
      [FIX]: Set the load balancer attribute 'load_balancing.cross_zone.enabled' to 'true'.
    >>
}
#
# Parameterized Rules
#
rule check(elastic_load_balancer) {
  %elastic_load_balancer[ Type == "network" ] {
    # Scenario 2
    LoadBalancerAttributes exists
    LoadBalancerAttributes is_list
    LoadBalancerAttributes not empty

    # Scenarios 3, 4 and 5
    some LoadBalancerAttributes[*] {
      Key exists
      Value exists

      Key == "load_balancing.cross_zone.enabled"
      Value in [ true, "true" ]
    }
  }
}
#
# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
CT.ELASTICLOADBALANCING.PR.14 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:

Vpc:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: 'true'
    EnableDnsHostnames: 'true'

SubnetOne:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: Vpc
    CidrBlock: 10.0.0.0/24
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: ''

SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: Vpc
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
        - 1
        - Fn::GetAZs: ''

NetworkLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Scheme: internal
    Type: network
    Subnets:
      - Ref: SubnetOne
      - Ref: SubnetTwo
    IpAddressType: ipv4
    LoadBalancerAttributes:
      - Key: load_balancing.cross_zone.enabled
        Value: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:

Vpc:
  Type: AWS::EC2::VPC
[CT.ELASTICLOADBALANCING.PR.15] Require that an Elastic Load Balancing v2 target group does not explicitly disable cross-zone load balancing

This control checks whether an Elastic Load Balancing v2 target group is configured so that it does not explicitly turn off cross-zone load balancing.

- **Control objective**: Improve availability
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::ElasticLoadBalancingV2::TargetGroup
- **AWS CloudFormation guard rule**: [CT.ELASTICLOADBALANCING.PR.15 rule specification](p. 824)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.ELASTICLOADBALANCING.PR.15 rule specification](p. 824)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.ELASTICLOADBALANCING.PR.15 example templates](p. 826)

**Explanation**

The nodes for your load balancer distribute requests from clients to registered targets. When cross-zone load balancing is enabled, each load balancer node distributes traffic across the registered targets in
all enabled Availability Zones. When cross-zone load balancing is not enabled, each load balancer node distributes traffic only across the registered targets in its Availability Zone.

The target group's cross-zone load balancing setting determines the load balancing behavior for the entire target group. For example, if cross-zone load balancing is enabled at the load balancer level, but not enabled at the target group level, traffic sent to the target group is not routed across Availability Zones.

**Usage considerations**

- This control checks only to ensure that cross-zone load balancing has not been explicitly turned off on an ELB target group. To ensure that cross-zone load balancing is enabled, be sure to enable this control in conjunction with related proactive controls that check load balancers directly.
- If you turn on cross-zone load balancing, you can't start a zonal shift. For more information, see [Resources supported for zonal shifts](https://docs.aws.amazon.com/r53recovery/latest/dg/arc-zonal-shift.resource-types.html) in the Amazon Route 53 Application Recovery Controller Developer Guide.

**Remediation for rule failure**

Do not set the load balancer attribute `load_balancing.cross_zone.enabled` to adopt the default value of `use_load_balancer_configuration`. Do not explicitly set the attribute to true, nor to the value `use_load_balancer_configuration`.

The examples that follow show how to implement this remediation.

**Elastic Load Balancer target group - Example**

Elastic Load Balancer target group configured to enable cross-zone load balancing. The example is shown in JSON and in YAML.

**JSON example**

```
{
    "TargetGroup": {
        "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
        "Properties": {
            "Protocol": "HTTP",
            "Port": 80,
            "VpcId": {
                "Ref": "VPC"
            },
            "TargetGroupAttributes": [
                {
                    "Key": "load_balancing.cross_zone.enabled",
                    "Value": true
                }
            ]
        }
    }
}
```

**YAML example**

```yaml
TargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
```
Properties:
   Protocol: HTTP
   Port: 80
   VpcId: !Ref 'VPC'

TargetGroupAttributes:
   - Key: load_balancing.cross_zone.enabled
     Value: true

CT.ELASTICLOADBALANCING.PR.15 rule specification

# ###########################################################################
# Rule Specification            #
###########################################################################
#
# Rule Identifier:
#   elbv2_target_group_cross_zone_check
#
# Description:
#   This control checks whether an Elastic Load Balancing v2 target group is configured so
#   that it does not explicitly turn off cross-zone load balancing.
#
# Reports on:
#   AWS::ElasticLoadBalancingV2::TargetGroup
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any Elastic Load Balancing v2 target group
#           resources
#     Then: SKIP
#
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an Elastic Load Balancing v2 target group resource
#     And: 'TargetGroupAttributes' has been provided as a non-empty list
#     And: 'TargetGroupAttributes' contain an entry with a 'Key' equal to
#          'load_balancing.cross_zone.enabled'
#          and 'Value' equal to a value other than bool(true), 'true' or
#          'use_load_balancer_configuration'
#     Then: FAIL
#
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an Elastic Load Balancing v2 target group resource
#     And: 'TargetGroupAttributes' has not been provided or has been provided as a list
#          that
does not contain an entry with a 'Key' equal to
#          'load_balancing.cross_zone.enabled'
#     Then: PASS
#
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an Elastic Load Balancing v2 target group resource
#     And: 'TargetGroupAttributes' has been provided as a non-empty list
And: 'TargetGroupAttributes' contains an entry with a 'Key' equal to 'load_balancing.cross_zone.enabled' and 'Value' equal to bool(true), string(true) or 'use_load_balancer_configuration'
Then: PASS

# Constants
let INPUT_DOCUMENT = this
let ELBV2_TARGET_GROUP_TYPE = "AWS::ElasticLoadBalancingV2::TargetGroup"
let ALLOWED_CROSS_ZONE_VALUES = ["true", true, "use_load_balancer_configuration"]

# Assignments
let elbv2_target_groups = Resources.*[ Type == %ELBV2_TARGET_GROUP_TYPE ]

# Primary Rules
# rule elbv2_target_group_cross_zone_check when is_cfn_template(%INPUT_DOCUMENT) %elbv2_target_groups not empty {
  check(%elbv2_target_groups.Properties)
  %elbv2_target_groups not empty {
    check(%elbv2_target_groups.Properties)
    [CT.ELASTICLOADBALANCING.PR.15]: Require that an Elastic Load Balancing v2 target
group does not explicitly disable cross-zone load balancing
    [FIX]: Do not set the load balancer attribute 'load_balancing.cross_zone.enabled'
to adopt the default value of 'use_load_balancer_configuration'. Do not explicitly set the
attribute to true, nor to the value 'use_load_balancer_configuration'.
  }
# rule elbv2_target_group_cross_zone_check when is_cfn_hook(%INPUT_DOCUMENT, %ELBV2_TARGET_GROUP_TYPE) {
  check(%INPUT_DOCUMENT.%ELBV2_TARGET_GROUP_TYPE.resourceProperties)
  [CT.ELASTICLOADBALANCING.PR.15]: Require that an Elastic Load Balancing v2 target
group does not explicitly disable cross-zone load balancing
    [FIX]: Do not set the load balancer attribute 'load_balancing.cross_zone.enabled'
to adopt the default value of 'use_load_balancer_configuration'. Do not explicitly set the
attribute to true, nor to the value 'use_load_balancer_configuration'.
#}
Value exists
Value in %ALLOWED_CROSS_ZONE_VALUES

# Utility Rules

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.ELASTICLOADBALANCING.PR.15 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: true
    EnableDnsHostnames: true
TargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    Protocol: HTTP
    Port: 80
    VpcId: Ref: VPC
    TargetGroupAttributes:
      - Key: load_balancing.cross_zone.enabled
        Value: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsSupport: true
    EnableDnsHostnames: true
TargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    Protocol: HTTP
Amazon Elastic Map Reduce (Amazon EMR) controls

Topics

- [CT.EMR.PR.1] Require that an Amazon Elastic MapReduce (EMR) security configuration is configured to encrypt data at rest in Amazon S3 (p. 827)
- [CT.EMR.PR.2] Require that an Amazon Elastic MapReduce (EMR) security configuration is configured to encrypt data at rest in Amazon S3 with an AWS KMS key (p. 832)
- [CT.EMR.PR.3] Require that an Amazon Elastic MapReduce (EMR) security configuration is configured with EBS volume local disk encryption using an AWS KMS key (p. 837)
- [CT.EMR.PR.4] Require that an Amazon Elastic MapReduce (EMR) security configuration is configured to encrypt data in transit (p. 842)

[CT.EMR.PR.1] Require that an Amazon Elastic MapReduce (EMR) security configuration is configured to encrypt data at rest in Amazon S3

This control checks whether an Amazon EMR security configuration is configured to encrypt EMR File System (EMRFS) objects at rest in Amazon S3.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EMR::SecurityConfiguration
- **AWS CloudFormation guard rule:** [CT.EMR.PR.1 rule specification](p. 828)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.EMR.PR.1 rule specification](p. 828)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.EMR.PR.1 example templates](p. 831)

Explanation

Amazon S3 encryption works with EMR File System (EMRFS) objects that are read from and written to Amazon S3. When you enable encryption at rest, you specify Amazon S3 server-side encryption (SSE) or client-side encryption (CSE) as the default encryption mode. Optionally, you can specify different encryption methods for individual buckets by using per bucket encryption overrides.

Remediation for rule failure

In the EncryptionConfiguration parameter, set the value of EnableAtRestEncryption to true, and provide an AtRestEncryptionConfiguration configuration.

The examples that follow show how to implement this remediation.
Amazon EMR security configuration - Example

An Amazon EMR security configuration configured to encrypt EMR File System (EMRFS) objects at rest in Amazon S3. The example is shown in JSON and in YAML.

JSON example

```
{
    "SecurityConfiguration": {
        "Type": "AWS::EMR::SecurityConfiguration",
        "Properties": {
            "SecurityConfiguration": {
                "EncryptionConfiguration": {
                    "EnableInTransitEncryption": false,
                    "EnableAtRestEncryption": true,
                    "AtRestEncryptionConfiguration": {
                        "S3EncryptionConfiguration": {
                            "EncryptionMode": "SSE-S3"
                        }
                    }
                }
            }
        }
    }
}
```

YAML example

```
SecurityConfiguration:
  Type: AWS::EMR::SecurityConfiguration
  Properties:
    SecurityConfiguration:
      EncryptionConfiguration:
        EnableInTransitEncryption: false
        EnableAtRestEncryption: true
        AtRestEncryptionConfiguration:
          S3EncryptionConfiguration:
            EncryptionMode: SSE-S3

CT.EMR.PR.1 rule specification

```
# #################################################################
##       Rule Specification       ##
# #################################################################
# # Rule Identifier:     # emr_sec_config_encryption_at_rest_s3_check #
# # Description:    # This control checks whether an Amazon EMR security configuration is configured to encrypt EMR File System (EMRFS) objects at rest in Amazon S3. #
# # Reports on:      # AWS::EMR::SecurityConfiguration #
# # Evaluates:     #
```

828
# AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document does not contain any EMR security configuration resources
# Then: SKIP
# Scenario: 2
# Given: The input document contains an EMR security configuration resource
# And: 'EncryptionConfiguration' in 'SecurityConfiguration' has not been provided
# Then: FAIL
# Scenario: 3
# Given: The input document contains an EMR security configuration resource
# And: 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a
# struct
# And: 'EnableAtRestEncryption' in 'EncryptionConfiguration' has not been provided
# or has been provided and set to a value other than bool(true)
# Then: FAIL
# Scenario: 4
# Given: The input document contains an EMR security configuration resource
# And: 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a
# struct
# And: 'EnableAtRestEncryption' in 'EncryptionConfiguration' has been provided and
# set to bool(true)
# And: 'AtRestEncryptionConfiguration' has not been provided
# Then: FAIL
# Scenario: 5
# Given: The input document contains an EMR security configuration resource
# And: 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a
# struct
# And: 'EnableAtRestEncryption' in 'EncryptionConfiguration' has been provided and
# set to bool(true)
# And: 'AtRestEncryptionConfiguration' has been provided as a struct
# And: 'EncryptionMode' in 'AtRestEncryptionConfiguration.S3EncryptionConfiguration'
# has not been provided or has been provided as an empty string
# Then: FAIL
# Scenario: 6
# Given: The input document contains an EMR security configuration resource
# And: 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a
# struct
# And: 'EnableAtRestEncryption' in 'EncryptionConfiguration' has been provided and
# set to bool(true)
# And: 'AtRestEncryptionConfiguration' has been provided as a struct
# And: 'EncryptionMode' in 'AtRestEncryptionConfiguration.S3EncryptionConfiguration'
# has been provided as a non-empty string
# Then: PASS

# Constants

let EMR_SECURITY_CONFIGURATION_TYPE = "AWS::EMR::SecurityConfiguration"
let INPUT_DOCUMENT = this
# Assignments

let emr_security_configurations = Resources.*[ Type == %EMR_SECURITY_CONFIGURATION_TYPE ]

# Primary Rules

rule emr_sec_config_encryption_at_rest_s3_check when is_cfn_template(%INPUT_DOCUMENT) %emr_security_configurations not empty
{
    check(%emr_security_configurations.Properties)
    <<
    [CT.EMR.PR.1]: Require that an Amazon Elastic MapReduce (EMR) security configuration is configured to encrypt data at rest in Amazon S3
    [FIX]: In the 'EncryptionConfiguration' parameter, set the value of 'EnableAtRestEncryption' to true, and provide an 'AtRestEncryptionConfiguration' configuration.
    >>
}

rule emr_sec_config_encryption_at_rest_s3_check when is_cfn_hook(%INPUT_DOCUMENT, %EMR_SECURITY_CONFIGURATION_TYPE) {
    check(%INPUT_DOCUMENT.%EMR_SECURITY_CONFIGURATION_TYPE.resourceProperties)
    <<
    [CT.EMR.PR.1]: Require that an Amazon Elastic MapReduce (EMR) security configuration is configured to encrypt data at rest in Amazon S3
    [FIX]: In the 'EncryptionConfiguration' parameter, set the value of 'EnableAtRestEncryption' to true, and provide an 'AtRestEncryptionConfiguration' configuration.
    >>
}

# Parameterized Rules

rule check(emr_security_configuration) {
    %emr_security_configuration {
        SecurityConfiguration exists
        SecurityConfiguration is_struct
        SecurityConfiguration {
            EncryptionConfiguration exists
            EncryptionConfiguration is_struct
            EncryptionConfiguration {
                EnableAtRestEncryption exists
                EnableAtRestEncryption == true
            }
            AtRestEncryptionConfiguration exists
            AtRestEncryptionConfiguration is_struct
            AtRestEncryptionConfiguration {
                S3EncryptionConfiguration exists
                S3EncryptionConfiguration is_struct
                S3EncryptionConfiguration {
                    EncryptionMode exists
                    check_is_string_and_not_empty(EncryptionMode)
                }
            }
        }
    }
}
CT.EMR.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  SecurityConfiguration:
    Type: AWS::EMR::SecurityConfiguration
    Properties:
      SecurityConfiguration:
        EncryptionConfiguration:
          EnableInTransitEncryption: false
          EnableAtRestEncryption: true
          AtRestEncryptionConfiguration:
            S3EncryptionConfiguration:
              EncryptionMode: SSE-S3
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
  SecurityConfiguration:
    Type: AWS::EMR::SecurityConfiguration
    Properties:
      SecurityConfiguration:
        EncryptionConfiguration:
          EnableAtRestEncryption: false
          EnableInTransitEncryption: false
[CT.EMR.PR.2] Require that an Amazon Elastic MapReduce (EMR) security configuration is configured to encrypt data at rest in Amazon S3 with an AWS KMS key

This control checks whether an Amazon EMR security configuration is configured to encrypt EMR File System (EMRFS) objects at rest in Amazon S3 with an AWS KMS key.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EMR::SecurityConfiguration
- **AWS CloudFormation guard rule:** CT.EMR.PR.2 rule specification (p. 833)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.EMR.PR.2 rule specification (p. 833)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.EMR.PR.2 example templates (p. 836)

**Explanation**

Amazon S3 encryption works with EMR File System (EMRFS) objects that are read from and written to Amazon S3. When you enable encryption at rest, you specify Amazon S3 server-side encryption (SSE) or client-side encryption (CSE) as the default encryption mode. Optionally, you can specify different encryption methods for individual buckets using per bucket encryption overrides.

**Remediation for rule failure**

In the EncryptionConfiguration parameter, set EnableAtRestEncryption to true, and provide an AtRestEncryptionConfiguration configuration, with EncryptionMode set to SSE-KMS or CSE-KMS.

The examples that follow show how to implement this remediation.

**Amazon EMR security configuration - Example**

An Amazon EMR security configuration configured to encrypt EMR File System (EMRFS) objects at rest in Amazon S3 with AWS KMS. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "SecurityConfiguration": {
        "Type": "AWS::EMR::SecurityConfiguration",
        "Properties": {
            "SecurityConfiguration": {
                "EncryptionConfiguration": {
                    "EnableInTransitEncryption": false,
                    "EnableAtRestEncryption": true,
                    "AtRestEncryptionConfiguration": {
                        "S3EncryptionConfiguration": {
                            "EncryptionMode": "SSE-KMS",
                            "AwsKmsKey": {
                                "Fn::GetAtt": [
                                    "SecurityConfiguration.AwsKmsKey", 
                                    "Ref", "bucket"
                                ]
                            }
                        }
                    }
                }
            }
        }
    }
}
```
YAML example

SecurityConfiguration:
  Type: AWS::EMR::SecurityConfiguration
  Properties:
    SecurityConfiguration:
      EncryptionConfiguration:
        EnableInTransitEncryption: false
        EnableAtRestEncryption: true
        AtRestEncryptionConfiguration:
          S3EncryptionConfiguration:
            EncryptionMode: SSE-KMS
            AwsKmsKey: !GetAtt 'KmsKey.Arn'

CT.EMR.PR.2 rule specification

# ###################################  
##       Rule Specification        
####################################
#  
# Rule Identifier:  
#   emr_sec_config_encryption_at_rest_s3_kms_check  
#  
# Description:  
#   This control checks whether an Amazon EMR security configuration is configured to encrypt EMR File System (EMRFS) objects at rest in Amazon S3 with an AWS KMS key.  
#  
# Reports on:  
#   AWS::EMR::SecurityConfiguration  
#  
# Evaluates:  
#   AWS CloudFormation, AWS CloudFormation hook  
#  
# Rule Parameters:  
#   None  
#  
# Scenarios:  
#   Scenario: 1  
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document  
#        And: The input document does not contain any EMR security configuration resources  
#        Then: SKIP  
#   Scenario: 2  
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document  
#        And: The input document contains an EMR security configuration resource
# Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- 'EncryptionConfiguration' has not been provided
- 'EnableAtRestEncryption' in 'EncryptionConfiguration' has not been provided or has been provided and set to a value other than bool(true)

# Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a struct
- 'EnableAtRestEncryption' in 'EncryptionConfiguration' has been provided and set to bool(true)
- 'AtRestEncryptionConfiguration' has not been provided

# Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a struct
- 'EnableAtRestEncryption' in 'EncryptionConfiguration' has been provided and set to bool(true)
- 'AtRestEncryptionConfiguration' has been provided as a struct
- 'EncryptionMode' in 'AtRestEncryptionConfiguration.S3EncryptionConfiguration' has not been provided or has been provided and set to a value other than a KMS-based encryption mode ('SSE-KMS', 'CSE-KMS')

# Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
- 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a struct
- 'EnableAtRestEncryption' in 'EncryptionConfiguration' has been provided and set to bool(true)
- 'AtRestEncryptionConfiguration' has been provided as a struct
- 'EncryptionMode' in 'AtRestEncryptionConfiguration.S3EncryptionConfiguration' has been provided and set to a KMS-based encryption mode ('SSE-KMS', 'CSE-KMS')
- 'Overrides' in 'AtRestEncryptionConfiguration.S3EncryptionConfiguration' has been provided as a non-empty list where one or more entries does not contain 'EncryptionMode', or contains 'EncryptionMode' set to a value other than a KMS-based encryption mode ('SSE-KMS', 'CSE-KMS')
And: 'Overrides' in 'AtRestEncryptionConfiguration.S3EncryptionConfiguration' has not been provided, or has been provided as an empty list, or list where every entry contains 'EncryptionMode' set to a KMS-based encryption mode ('SSE-KMS', 'CSE-KMS')
Then: PASS

Constants

let EMR_SECURITY_CONFIGURATION_TYPE = "AWS::EMR::SecurityConfiguration"
let S3_KMS_ENCRYPTION_MODES = [ "SSE-KMS", "CSE-KMS" ]
let INPUT_DOCUMENT = this

Assignments

let emr_security_configurations = Resources.*[ Type == %EMR_SECURITY_CONFIGURATION_TYPE ]

Primary Rules

rule emr_sec_config_encryption_at_rest_s3_kms_check when is_cfn_template(%INPUT_DOCUMENT)
%emr_security_configurations not empty {
    check(%emr_security_configurations.Properties)
    <<
    [CT.EMR.PR.2]: Require that an Amazon Elastic MapReduce (EMR) security configuration is configured to encrypt data at rest in Amazon S3 with an AWS KMS key
    [FIX]: In the 'EncryptionConfiguration' parameter, set 'EnableAtRestEncryption' to true, and provide an 'AtRestEncryptionConfiguration' configuration, with 'EncryptionMode' set to 'SSE-KMS' or 'CSE-KMS'.
    >>
}

rule emr_sec_config_encryption_at_rest_s3_kms_check when is_cfn_hook(%INPUT_DOCUMENT,%EMR_SECURITY_CONFIGURATION_TYPE) {
    check(%INPUT_DOCUMENT.%EMR_SECURITY_CONFIGURATION_TYPE.resourceProperties)
    <<
    [CT.EMR.PR.2]: Require that an Amazon Elastic MapReduce (EMR) security configuration is configured to encrypt data at rest in Amazon S3 with an AWS KMS key
    [FIX]: In the 'EncryptionConfiguration' parameter, set 'EnableAtRestEncryption' to true, and provide an 'AtRestEncryptionConfiguration' configuration, with 'EncryptionMode' set to 'SSE-KMS' or 'CSE-KMS'.
    >>
}

Parameterized Rules

rule check(emr_security_configuration) {
    %emr_security_configuration {
        SecurityConfiguration exists
        SecurityConfiguration is_struct

        SecurityConfiguration {
            # Scenario 2
            EncryptionConfiguration exists
            EncryptionConfiguration is_struct

            EncryptionConfiguration {
                # Scenario 3
                EnableAtRestEncryption exists
                EnableAtRestEncryption == true
            }
        }
    }
}
# Scenario 4
AtRestEncryptionConfiguration exists
AtRestEncryptionConfiguration is_struct

# Scenarios 5, 6 and 7
AtRestEncryptionConfiguration {
  S3EncryptionConfiguration exists
  S3EncryptionConfiguration is_struct

  let s3_encryption_configuration = S3EncryptionConfiguration

  %s3_encryption_configuration {
    check_kms_key_configuration(this)
  }

  %s3_encryption_configuration [
    Overrides exists
    Overrides is_list
    Overrides not empty
  ] {
    Overrides[*] {
      check_kms_key_configuration(this)
    }
  }
}
}
}
}
}
}

rule check_kms_key_configuration(s3_encryption_config) {
  %s3_encryption_config {
    EncryptionMode exists
    EncryptionMode in %S3_KMS_ENCRYPTION_MODES
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.EMR.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  KmsKey:
    Type: AWS::KMS::Key
    Properties:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
SecurityConfiguration:
  Type: AWS::EMR::SecurityConfiguration
Properties:
  SecurityConfiguration:
    EncryptionConfiguration:
      EnableIn TransitEncryption: false
      EnableAtRestEncryption: false
      AtRestEncryptionConfiguration:
        S3EncryptionConfiguration:
          EncryptionMode: SSE-KMS
          AwsKmsKey:
            Fn::GetAtt:
              - KmsKey
              - Arn

[CT.EMR.PR.3] Require that an Amazon Elastic MapReduce (EMR) security configuration is configured with EBS volume local disk encryption using an AWS KMS key

This control checks whether Amazon EMR security configurations are configured with local disk encryption enabled, using EBS volume encryption and AWS KMS.

- Control objective: Encrypt data at rest
- Implementation: AWS CloudFormation guard rule
- Control behavior: Proactive
- Resource types: AWS::EMR::SecurityConfiguration
- AWS CloudFormation guard rule: CT.EMR.PR.3 rule specification (p. 839)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.EMR.PR.3 rule specification (p. 839)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.EMR.PR.3 example templates (p. 841)

Explanation

For data at rest, EMR provides the option to encrypt local disk storage. The local storage consists of EC2 instance store volumes and the attached Amazon Elastic Block Store (EBS) storage, which are provisioned with your cluster. The EBS encryption option encrypts the EBS root device volume and its attached storage volumes. The EBS encryption option is available only when you specify AWS Key Management Service as your key provider. AWS Control Tower recommends using EBS encryption.

Usage considerations

• If you create a cluster in a Region where Amazon EC2 encryption of EBS volumes is enabled by default for your account, an EBS volume is encrypted even when local disk encryption is not enabled.
• With local disk encryption enabled in a security configuration, the Amazon EMR settings take precedence over the Amazon EC2 encryption-by-default settings for cluster EC2 instances.

Remediation for rule failure

In the EncryptionConfiguration parameter, set the value of EnableAtRestEncryption to true, and provide an AtRestEncryptionConfiguration configuration, containing an LocalDiskEncryptionConfiguration configuration that sets EnableEbsEncryption to true.

The examples that follow show how to implement this remediation.

Amazon EMR security configuration - Example

An Amazon EMR security configuration configured with EBS encryption using AWS KMS. The example is shown in JSON and in YAML.

JSON example

```json
{
  "SecurityConfiguration": {
    "Type": "AWS::EMR::SecurityConfiguration",
    "Properties": {
      "SecurityConfiguration": {
        "EncryptionConfiguration": {
          "EnableInTransitEncryption": false,
          "EnableAtRestEncryption": true,
          "AtRestEncryptionConfiguration": {
            "LocalDiskEncryptionConfiguration": {
              "EnableEbsEncryption": true,
              "EncryptionKeyProviderType": "AwsKms",
              "AwsKmsKey": "arn:aws:kms:us-west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            }
          }
        }
      }
    }
  }
}
```

YAML example

```yaml
838
```
SecurityConfiguration:
  Type: AWS::EMR::SecurityConfiguration
  Properties:
    SecurityConfiguration:
      EncryptionConfiguration:
        EnableInTransitEncryption: false
        EnableAtRestEncryption: true
        AtRestEncryptionConfiguration:
          LocalDiskEncryptionConfiguration:
            EnableEbsEncryption: true
            EncryptionKeyProviderType: AwsKms
            AwsKmsKey: arn:aws:kms:us-west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab

CT.EMR.PR.3 rule specification

# ####################################################################
##       Rule Specification        ##
# ####################################################################
#
# Rule Identifier:
#   emr_sec_config_ebs_encryption_check
#
# Description:
#   This control checks whether Amazon EMR security configurations are configured with
#   local disk encryption enabled, using EBS volume encryption and AWS KMS.
#
# Reports on:
#   AWS::EMR::SecurityConfiguration
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any EMR security configuration resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EMR security configuration resource
#     And: 'EncryptionConfiguration' in 'SecurityConfiguration' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an EMR security configuration resource
#     And: 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a
#       struct
#     And: 'EnableAtRestEncryption' in 'EncryptionConfiguration' has not been provided
#       or has been provided and set to a value other than bool(true)
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document contains an EMR security configuration resource
And: 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a struct
And: 'EnableAtRestEncryption' in 'EncryptionConfiguration' has been provided and set to bool(true)
And: 'AtRestEncryptionConfiguration' has not been provided
Then: FAIL
Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an EMR security configuration resource
And: 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a struct
And: 'EnableAtRestEncryption' in 'EncryptionConfiguration' has been provided and set to bool(true)
And: 'AtRestEncryptionConfiguration' has been provided as a struct
And: 'EnableEbsEncryption' in 'AtRestEncryptionConfiguration.LocalDiskEncryptionConfiguration'
has not been provided or has been provided and set to a value other than bool(true)
Then: FAIL
Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an EMR security configuration resource
And: 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a struct
And: 'EnableAtRestEncryption' in 'EncryptionConfiguration' has been provided and set to bool(true)
And: 'AtRestEncryptionConfiguration' has been provided as a struct
And: 'EnableEbsEncryption' in 'AtRestEncryptionConfiguration.LocalDiskEncryptionConfiguration'
has been provided and set to bool(true)
Then: PASS

# Constants
let EMR_SECURITY_CONFIGURATION_TYPE = "AWS::EMR::SecurityConfiguration"
let INPUT_DOCUMENT = this

# Assignments
let emr_security_configurations = Resources.*[ Type == %EMR_SECURITY_CONFIGURATION_TYPE ]

# Primary Rules
rule emr_sec_config_ebs_encryption_check when is_cfn_template(%INPUT_DOCUMENT)
%emr_security_configurations not empty {
  check(%emr_security_configurations.Properties)
  <<<
  [CT.EMR.PR.3]: Require that an Amazon Elastic MapReduce (EMR) security configuration is configured with EBS volume local disk encryption using an AWS KMS key
  [FIX]: In the 'EncryptionConfiguration' parameter, set the value of 'EnableAtRestEncryption' to true, and provide an 'AtRestEncryptionConfiguration' configuration, containing an 'LocalDiskEncryptionConfiguration' configuration that sets 'EnableEbsEncryption' to true.
  >>>
}
rule emr_sec_config_ebs_encryption_check when is_cfn_hook(%INPUT_DOCUMENT, %EMR_SECURITY_CONFIGURATION_TYPE) {
  check(%INPUT_DOCUMENT.%EMR_SECURITY_CONFIGURATION_TYPE.resourceProperties)
  <<<

840
[CT.EMR.PR.3]: Require that an Amazon Elastic MapReduce (EMR) security configuration is configured with EBS volume local disk encryption using an AWS KMS key

[FIX]: In the 'EncryptionConfiguration' parameter, set the value of 'EnableAtRestEncryption' to true, and provide an 'AtRestEncryptionConfiguration' configuration, containing an 'LocalDiskEncryptionConfiguration' configuration that sets 'EnableEbsEncryption' to true.

```plaintext
# Parameterized Rules
#
rule check(emr_security_configuration) {
  %emr_security_configuration {
    SecurityConfiguration exists
    SecurityConfiguration is_struct
      SecurityConfiguration {
        # Scenario 2
        EncryptionConfiguration exists
        EncryptionConfiguration is_struct
          EncryptionConfiguration {
            # Scenario 3
            EnableAtRestEncryption exists
            EnableAtRestEncryption == true

            # Scenario 4
            AtRestEncryptionConfiguration exists
            AtRestEncryptionConfiguration is_struct
              AtRestEncryptionConfiguration {
                LocalDiskEncryptionConfiguration exists
                LocalDiskEncryptionConfiguration is_struct
                  LocalDiskEncryptionConfiguration {
                    EnableEbsEncryption exists
                    EnableEbsEncryption == true
                  }
            }
          }
        }
      }
  }
}
#
# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

**CT.EMR.PR.3 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.
PASS Example - Use this template to verify a compliant resource creation.

<table>
<thead>
<tr>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>KmsKey:</td>
</tr>
<tr>
<td>Type: AWS::KMS::Key</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>KeyPolicy:</td>
</tr>
<tr>
<td>Version: 2012-10-17</td>
</tr>
<tr>
<td>Id: example-key-policy</td>
</tr>
<tr>
<td>Statement:</td>
</tr>
<tr>
<td>- Sid: Enable IAM User Permissions</td>
</tr>
<tr>
<td>Effect: Allow</td>
</tr>
<tr>
<td>Principal:</td>
</tr>
<tr>
<td>AWS:</td>
</tr>
<tr>
<td>Fn::Sub: arn:${AWS::Partition}:iam::${AWS::AccountId}:root</td>
</tr>
<tr>
<td>Action: kms:*</td>
</tr>
<tr>
<td>Resource: &quot;*&quot;</td>
</tr>
</tbody>
</table>

| SecurityConfiguration: |
| Type: AWS::EMR::SecurityConfiguration |
| Properties: |
| SecurityConfiguration: |
| EncryptionConfiguration: |
| EnableIn TransitEncryption: false |
| EnableAtRestEncryption: true |
| AtRestEncryptionConfiguration: |
| LocalDiskEncryptionConfiguration: |
| EnableEbsEncryption: true |
| EncryptionKeyIdProviderType: AwsKms |
| AwsKmsKey: |
| Fn::GetAtt: |
| - KmsKey |
| - Arn |

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

| Resources: |
| SecurityConfiguration: |
| Type: AWS::EMR::SecurityConfiguration |
| Properties: |
| SecurityConfiguration: |
| EncryptionConfiguration: |
| EnableIn TransitEncryption: false |
| EnableAtRestEncryption: false |

[CT.EMR.PR.4] Require that an Amazon Elastic MapReduce (EMR) security configuration is configured to encrypt data in transit

This control checks whether an Amazon EMR security configuration is configured to require encryption in transit.

- **Control objective:** Encrypt data in transit
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::EMR::SecurityConfiguration
• **AWS CloudFormation guard rule**: [CT.EMR.PR.4 rule specification (p. 844)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: [CT.EMR.PR.4 rule specification (p. 844)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.EMR.PR.4 example templates (p. 847)]

**Explanation**

For data in transit, EMR security configurations provide you two options. You can create PEM certificates, zip them in a file, and reference them from Amazon S3, or you can implement a certificate custom provider in Java and specify the S3 path to the JAR. In either case, EMR downloads artifacts to each node in the cluster automatically, and later uses them to implement open-source, in-transit encryption features. For more information on how these certificates are used with different big data technologies, see Amazon EMR documentation.

**Usage considerations**

- Several encryption mechanisms are associated with in-transit encryption. These mechanisms are open-source features, they are application-specific, and they may vary by Amazon EMR release. For more information, see Encryption in transit in the Amazon EMR Management Guide.

**Remediation for rule failure**

In the EncryptionConfiguration parameter, set the EnableInTransitEncryption parameter to true, and provide an InTransitEncryptionConfiguration configuration.

The examples that follow show how to implement this remediation.

**Amazon EMR security configuration - Example**

An Amazon EMR security configuration configured to require encryption of data in transit. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "SecurityConfiguration": {
    "Type": "AWS::EMR::SecurityConfiguration",
    "Properties": {
      "SecurityConfiguration": {
        "EncryptionConfiguration": {
          "EnableAtRestEncryption": false,
          "EnableInTransitEncryption": true,
          "InTransitEncryptionConfiguration": {
            "TLSCertificateConfiguration": {
              "CertificateProviderType": "PEM",
              "S3Object": "s3://MyConfigStore/artifacts/MyCerts.zip"
            }
          }
        }
      }
    }
  }
}
```
YAML example

SecurityConfiguration:
  Type: AWS::EMR::SecurityConfiguration
  Properties:
    SecurityConfiguration:
      EncryptionConfiguration:
        EnableAtRestEncryption: false
        EnableInTransitEncryption: true
        InTransitEncryptionConfiguration:
          TLSCertificateConfiguration:
            CertificateProviderType: PEM
            S3Object: s3://MyConfigStore/artifacts/MyCerts.zip

CT.EMR.PR.4 rule specification

# # Rule Specification #
# Rule Identifier: #
#   emr_sec_config_encryption_in_transit_check #
# Description: #
# This control checks whether an Amazon EMR security configuration is configured to #
# require encryption in transit. #
# Reports on: #
#   AWS::EMR::SecurityConfiguration #
# Evaluates: #
#   AWS CloudFormation, AWS CloudFormation hook #
# Rule Parameters: #
#   None #
# Scenarios: #
# Scenario: 1 #
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook #
#       document #
#       And: The input document does not contain any EMR security configuration resources #
#       Then: SKIP #
# Scenario: 2 #
#   Given: The input document contains an EMR security configuration resource #
#       And: 'EncryptionConfiguration' in 'SecurityConfiguration' has not been provided #
#       Then: FAIL #
# Scenario: 3 #
#   Given: The input document contains an EMR security configuration resource #
#       And: 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a #
#       struct #
#       And: 'EnableInTransitEncryption' in 'EncryptionConfiguration' has not been provided #
#       or has been provided and set to a value other than bool(true) #
#       Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EMR security configuration resource
# And: 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a struct
# And: 'EnableInTransitEncryption' in 'EncryptionConfiguration' has been provided and set to bool(true)
# And: 'InTransitEncryptionConfiguration' has not been provided
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EMR security configuration resource
# And: 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a struct
# And: 'EnableInTransitEncryption' in 'EncryptionConfiguration' has been provided and set to bool(true)
# And: 'InTransitEncryptionConfiguration' has been provided as a struct
# And: 'CertificateProviderType' in 'InTransitEncryptionConfiguration.TLSCertificateConfiguration' has not been provided or has been provided and set to an empty string
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an EMR security configuration resource
# And: 'EncryptionConfiguration' in 'SecurityConfiguration' has been provided as a struct
# And: 'EnableInTransitEncryption' in 'EncryptionConfiguration' has been provided and set to bool(true)
# And: 'InTransitEncryptionConfiguration' has been provided as a struct
# And: 'CertificateProviderType' in 'InTransitEncryptionConfiguration.TLSCertificateConfiguration' has been provided and set to a non-empty string
# Then: PASS

# Constants

let EMR_SECURITY_CONFIGURATION_TYPE = "AWS::EMR::SecurityConfiguration"
let INPUT_DOCUMENT = this

# Assignments

let emr_security_configurations = Resources.*[ Type == %EMR_SECURITY_CONFIGURATION_TYPE ]

# Primary Rules

rule emr_sec_config_encryption_in_transit_check when is_cfn_template(%INPUT_DOCUMENT)
    %emr_security_configurations not empty
    {
        check(%emr_security_configurations.Properties)
        <<
        [CT.EMR.PR.4]: Require that an Amazon Elastic MapReduce (EMR) security configuration is configured to encrypt data in transit
        [FIX]: In the 'EncryptionConfiguration' parameter, set the 'EnableInTransitEncryption' parameter to true, and provide an 'InTransitEncryptionConfiguration' configuration.
        >>
    }

rule emr_sec_config_encryption_in_transit_check when is_cfn_hook(%INPUT_DOCUMENT,
    %EMR_SECURITY_CONFIGURATION_TYPE) {
check(%INPUT_DOCUMENT.%EMR_SECURITY_CONFIGURATION_TYPE.resourceProperties)
<<
[CT.EMR.PR.4]: Require that an Amazon Elastic MapReduce (EMR) security configuration is configured to encrypt data in transit
[FIX]: In the 'EncryptionConfiguration' parameter, set the 'EnableInTransitEncryption' parameter to true, and provide an 'InTransitEncryptionConfiguration' configuration.
>>

# Parameterized Rules
#
rule check(emr_security_configuration) {
  %emr_security_configuration {
    SecurityConfiguration exists
    SecurityConfiguration is_struct
    SecurityConfiguration {
      # Scenario 2
      EncryptionConfiguration exists
      EncryptionConfiguration is_struct
      EncryptionConfiguration {
        # Scenario 3
        EnableInTransitEncryption exists
        EnableInTransitEncryption == true
        # Scenario 4
        InTransitEncryptionConfiguration exists
        InTransitEncryptionConfiguration is_struct
        # Scenarios 5 and 6
        InTransitEncryptionConfiguration {
          TLSCertificateConfiguration exists
          TLSCertificateConfiguration is_struct
          TLSCertificateConfiguration {
            CertificateProviderType exists
            check_is_string_and_not_empty(CertificateProviderType)
          }
        }
      }
    }
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists  or
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
rule check_is_string_and_not_empty(value) {
  %value {
    this is_string
    this != /\A\s*\z/
CT.EMR.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
- SecurityConfiguration:
  - Type: AWS::EMR::SecurityConfiguration
  - Properties:
    - SecurityConfiguration:
      - EncryptionConfiguration:
        - EnableAtRestEncryption: false
        - EnableInTransitEncryption: true
        - InTransitEncryptionConfiguration:
          - TLSCertificateConfiguration:
            - CertificateProviderType: PEM
            - S3Object: s3://MyConfigStore/artifacts/MyCerts.zip

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
- SecurityConfiguration:
  - Type: AWS::EMR::SecurityConfiguration
  - Properties:
    - SecurityConfiguration:
      - EncryptionConfiguration:
        - EnableInTransitEncryption: false
        - EnableAtRestEncryption: false

AWS Glue controls

Topics
- [CT.GLUE.PR.1] Require an AWS Glue job to have an associated security configuration (p. 847)

[CT.GLUE.PR.1] Require an AWS Glue job to have an associated security configuration

This control checks whether an AWS Glue job has an associated security configuration.

- Control objective: Encrypt data at rest
- Implementation: AWS CloudFormation guard rule
- Control behavior: Proactive
- Resource types: AWS::Glue::Job
- AWS CloudFormation guard rule: CT.GLUE.PR.1 rule specification (p. 849)

Details and examples
Proactive controls

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.GLUE.PR.1 rule specification (p. 849)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.GLUE.PR.1 example templates (p. 851)]

Explanation

In AWS Glue, a security configuration contains the properties that are needed when you write encrypted data. Security configurations for an AWS Glue job must be configured to specify how the data is encrypted at the Amazon S3 target. Encryption helps protect the data from unauthorized access and exposure.

Remediation for rule failure

Set the SecurityConfiguration parameter to the name of an AWS Glue security configuration.

The examples that follow show how to implement this remediation.

AWS Glue job - Example

An AWS Glue job configured with an associated security configuration. The example is shown in JSON and in YAML.

JSON example

```json
{
    "GlueJob": {
        "Type": "AWS::Glue::Job",
        "Properties": {
            "Command": {
                "Name": "glueetl",
                "ScriptLocation": "s3://example-glue-script-bucket/scripts"
            },
            "Name": "sample-glue-job",
            "Role": {
                "Ref": "IAMRole"
            },
            "GlueVersion": "2.0",
            "SecurityConfiguration": "GlueSecurityConfig"
        }
    }
}
```

YAML example

```yaml
GlueJob:
  Type: AWS::Glue::Job
  Properties:
    Command:
      Name: glueetl
      ScriptLocation: s3://example-glue-script-bucket/scripts
    Name: sample-glue-job
    Role: !Ref 'IAMRole'
    GlueVersion: '2.0'
    SecurityConfiguration: !Ref 'GlueSecurityConfig'
```
CT.GLUE.PR.1 rule specification

---

# Rule Specification
## Rule Identifier:
- glue_job_security_config_check

## Description:
- This control checks whether an AWS Glue job has an associated security configuration.

## Reports on:
- AWS::Glue::Job

## Evaluates:
- AWS CloudFormation, AWS CloudFormation hook

## Rule Parameters:
- None

## Scenarios:
1. **Scenario: 1**
   - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   - And: The input document does not contain any AWS Glue job resources
   - Then: SKIP

2. **Scenario: 2**
   - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   - And: The input document contains an AWS Glue job resource
   - And: 'SecurityConfiguration' has not been provided
   - Then: FAIL

3. **Scenario: 3**
   - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   - And: The input document contains an AWS Glue job resource
   - And: 'SecurityConfiguration' has been provided as an empty string or invalid local reference
   - Then: FAIL

4. **Scenario: 4**
   - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   - And: The input document contains an AWS Glue job resource
   - And: 'SecurityConfiguration' has been provided as a non-empty string or valid local reference to an AWS Glue security configuration resource
   - Then: PASS

## Constants
- INPUT_DOCUMENT = this
- GLUE_JOB_TYPE = "AWS::Glue::Job"

## Assignments
- glue_jobs = Resources.*[ Type == %GLUE_JOB_TYPE ]

## Primary Rules
---
## Proactive controls

### Rule: Proactive controls - Proactive controls

1. **Rule: glue_job_security_config_check**
   - When: is_cfn_template(%INPUT_DOCUMENT)
   - Check: %glue_jobs not empty
     - **[CT.GLUE.PR.1]**: Require an AWS Glue job to have an associated security configuration
     - **[FIX]**: Set the 'SecurityConfiguration' parameter to the name of an AWS Glue security configuration.

2. **Rule: glue_job_security_config_check**
   - When: is_cfn_hook(%INPUT_DOCUMENT, %GLUE_JOB_TYPE)
   - Check: %INPUT_DOCUMENT.%GLUE_JOB_TYPE.resourceProperties
     - **[CT.GLUE.PR.1]**: Require an AWS Glue job to have an associated security configuration
     - **[FIX]**: Set the 'SecurityConfiguration' parameter to the name of an AWS Glue security configuration.

### Parameterized Rules

1. **Rule: check(glue_job)**
   - %glue_job
     - **Scenario 2:** SecurityConfiguration exists
     - **Scenario 3 and 4:**
       - check_is_string_and_not_empty(SecurityConfiguration) or
       - check_local_references(%INPUT_DOCUMENT, SecurityConfiguration, "AWS::Glue::SecurityConfiguration")

### Utility Rules

1. **Rule: check_is_string_and_not_empty(value)**
   - %value
     - **this is_string**
     - **this != \A\s*\z/**

2. **Rule: is_cfn_template(doc)**
   - %doc
     - **AWSTemplateFormatVersion exists** or
     - **Resources exists**

3. **Rule: is_cfn_hook(doc, RESOURCE_TYPE)**
   - %doc.%RESOURCE_TYPE.resourceProperties exists

4. **Rule: check_local_references(doc, reference_properties, referenced_RESOURCE_TYPE)**
   - %reference_properties
     - **'Fn::GetAtt'**
       - query_for_resource(%doc, this[0], %referenced_RESOURCE_TYPE)
       - **Local Stack reference was invalid**
     - **Ref**
       - query_for_resource(%doc, this, %referenced_RESOURCE_TYPE)
       - **Local Stack reference was invalid**
rule query_for_resource(doc, resource_key, referenced_RESOURCE_TYPE) {
  let referenced_resource = %doc.Resources[ keys == %resource_key ]
  %referenced_resource not empty
  %referenced_resource {
    Type == %referenced_RESOURCE_TYPE
  }
}

**CT.GLUE.PR.1 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```json
Resources:
  IAMRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: "Allow"
            Principal:
              Service:
                - "glue.amazonaws.com"
            Action:
              - "sts:AssumeRole"
            Path: "/"
  Key:
    Type: AWS::KMS::Key
    Properties:
      KeyPolicy:
        Version: 2012-10-17
        Id: example-policy
        Statement:
          - Sid: Enable IAM User Permissions
            Effect: Allow
            Principal:
              AWS:
                Fn::Sub: arn:${AWS::Partition}:iam::${AWS::AccountId}:root
            Action: kms:*
            Resource: "*"
        KeySpec: SYMMETRIC_DEFAULT
        EnableKeyRotation: true
  GlueSecurityConfig:
    Type: AWS::Glue::SecurityConfiguration
    Properties:
      Name:
        Fn::Sub: ${AWS::StackName}-example
      EncryptionConfiguration:
        SSEEncryptions:
          - KmsKeyArn:
            Fn::GetAtt: [Key, Arn]
          SSEEncryptionMode: SSE-KMS
  GlueJob:
    Type: AWS::Glue::Job
    Properties:
      Command:
```

851
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
IAMRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: "Allow"
          Principal:
            Service:
              - "glue.amazonaws.com"
          Action:
            - "sts:AssumeRole"
          Path: "/
GlueJob:
  Type: AWS::Glue::Job
  Properties:
    Command:
      Name: glueetl
      ScriptLocation: "s3://example-glue-script-bucket/scripts"
    Name:
      Fn::Sub: ${AWS::StackName}-example
    Role:
      Ref: IAMRole
    GlueVersion: "2.0"

Amazon GuardDuty controls

Topics
- [CT.GUARDDUTY.PR.1] Require an Amazon GuardDuty detector to have Amazon S3 protection activated (p. 852)

[CT.GUARDDUTY.PR.1] Require an Amazon GuardDuty detector to have Amazon S3 protection activated

This control checks whether Amazon S3 protection is enabled on an Amazon GuardDuty detector.

- **Control objective**: Establish logging and monitoring
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::GuardDuty::Detector
- **AWS CloudFormation guard rule**: [CT.GUARDDUTY.PR.1 rule specification (p. 854)](https://docs.aws.amazon.com/ct/latest/userguide/proactive.html#ct.guarrdduty.pr.1)
Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: CT.GUARDDUTY.PR.1 rule specification (p. 854)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.GUARDDUTY.PR.1 example templates (p. 856)

Explanation

Amazon GuardDuty monitors threats against your Amazon S3 resources by analyzing AWS CloudTrail management events and CloudTrail S3 data events. These data sources monitor different kinds of activity. For example, CloudTrail management events for S3 include operations that list or configure S3 buckets, such as ListBuckets, DeleteBuckets, and PutBucketReplication. Examples of data events for S3 include object-level API operations, such as GetObject, ListObjects, DeleteObject, and PutObject.

Amazon GuardDuty monitoring of AWS CloudTrail management events is on by default for all accounts that have enabled GuardDuty, and it is not configurable. Amazon S3 data event logs are a configurable data source in GuardDuty.

AWS Control Tower recommends that you enable Amazon S3 protection in GuardDuty. If the feature is not enabled, GuardDuty cannot fully monitor or generate findings for suspicious access to data stored in your Amazon S3 buckets.

Remediation for rule failure

Set DataSources.S3Logs to true.

The examples that follow show how to implement this remediation.

GuardDuty Detector - Example

Amazon GuardDuty detector with Amazon S3 protection enabled. The example is shown in JSON and in YAML.

JSON example

```json
{
  "GuardDutyDetector": {
    "Type": "AWS::GuardDuty::Detector",
    "Properties": {
      "Enable": true,
      "DataSources": {
        "S3Logs": {
          "Enable": true
        }
      }
    }
  }
}
```

YAML example

```yaml
GuardDutyDetector:
  Type: AWS::GuardDuty::Detector
  Properties:
    DataSources:
      S3Logs:
        Enable: true
```

853
Enable: true
DataSources:
  S3Logs:
    Enable: true

CT.GUARDDUTY.PR.1 rule specification

# ####################################################################
##       Rule Specification        ##
#####################################################################
#
# Rule Identifier:
#   guardduty_s3_protection_enabled_check
#
# Description:
#   Checks if Amazon S3 protection is enabled on an Amazon GuardDuty detector.
#
# Reports on:
#   AWS::GuardDuty::Detector
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#      Given: The input document is an AWS CloudFormation or CloudFormation hook document
#      And: The input document does not contain any Amazon GuardDuty detector resources
#      Then: SKIP
#
#   Scenario: 2
#      Given: The input document is an AWS CloudFormation or CloudFormation hook document
#      And: The input document contains a GuardDuty detector resource
#      And: 'Enable' has not been specified or specified with value is bool(false)
#      Then: FAIL
#
#   Scenario: 3
#      Given: The input document is an AWS CloudFormation or CloudFormation hook document
#      And: The input document contains a GuardDuty detector resource
#      And: 'Enable' is specified with a value of bool(true)
#      And: 'DataSources.S3Logs' has not been specified
#      Then: FAIL
#
#   Scenario: 4
#      Given: The input document is an AWS CloudFormation or CloudFormation hook document
#      And: The input document contains a GuardDuty detector resource
#      And: 'Enable' is specified and value is bool(true)
#      And: 'DataSources.S3Logs' has been specified
#      And: 'Enable' has not been specified within 'S3Logs' or has been specified with a value of bool(false)
#      Then: FAIL
#
#   Scenario: 5
#      Given: The input document is an AWS CloudFormation or CloudFormation hook document
#      And: The input document contains a GuardDuty detector resource
#      And: 'Enable' is specified and value is bool(true)
#      And: 'DataSources.S3Logs' has been specified
#      And: 'Enable' has been specified within 'S3Logs' with a value of bool(true)
#      Then: PASS
#
# Constants
#
let GUARDDUTY_DETECTOR_TYPE = "AWS::GuardDuty::Detector"
let INPUT_DOCUMENT = this
#
# Assignments
#
let guardduty_detectors = Resources.*[ Type == %GUARDDUTY_DETECTOR_TYPE ]
#
# Primary Rules
#
rule guardduty_s3_protection_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
  %guardduty_detectors not empty {
    check(%guardduty_detectors.Properties)
    <<
    [CT.GUARDDUTY.PR.1]: Require an Amazon GuardDuty detector to have Amazon S3 protection activated
    [FIX]: Set 'DataSources.S3Logs' to true.
    >>
  }
rule guardduty_s3_protection_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %GUARDDUTY_DETECTOR_TYPE) {
  check(%INPUT_DOCUMENT.%GUARDDUTY_DETECTOR_TYPE.resourceProperties)
  <<
  [CT.GUARDDUTY.PR.1]: Require an Amazon GuardDuty detector to have Amazon S3 protection activated
  [FIX]: Set 'DataSources.S3Logs' to true.
  >>
}
#
# Parameterized Rules
#
rule check(guardduty_detector) {
  %guardduty_detector {
    # Scenario: 2
    Enable exists
    Enable == true
    # Scenario: 3
    DataSources exists
    DataSources is_struct
    DataSources {
      # Scenario: 4 and 5
      S3Logs exists
      S3Logs is_struct
      S3Logs {
        Enable exists
        Enable == true
      }
    }
  }
}
#
# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.GUARDDUTY.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
GuardDutyDetector:
  Type: AWS::GuardDuty::Detector
  Properties:
    Enable: true
    DataSources:
      S3Logs:
        Enable: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
GuardDutyDetector:
  Type: AWS::GuardDuty::Detector
  Properties:
    Enable: true
    DataSources:
      S3Logs:
        Enable: false

AWS Identity and Access Management (IAM) controls

Topics
- [CT.IAM.PR.1] Require that an AWS Identity and Access Management (IAM) inline policy does not have a statement that includes "*" in the Action and Resource elements (p. 856)
- [CT.IAM.PR.2] Require that AWS Identity and Access Management (IAM) customer-managed policies do not contain a statement that includes "*" in the Action and Resource elements (p. 865)
- [CT.IAM.PR.3] Require that AWS Identity and Access Management (IAM) customer-managed policies do not have wildcard service actions (p. 870)
- [CT.IAM.PR.4] Require that an AWS Identity and Access Management (IAM) user does not have an inline or managed policy attached (p. 876)
- [CT.IAM.PR.5] Require that AWS Identity and Access Management (IAM) inline policies do not have wildcard service actions (p. 882)

[CT.IAM.PR.1] Require that an AWS Identity and Access Management (IAM) inline policy does not have a statement that includes "*" in the Action and Resource elements

This control checks that AWS Identity and Access Management (IAM) inline policies do not include
Effect: Allow with Action: * over Resource: *.

856
• **Control objective:** Enforce least privilege
• **Implementation:** AWS CloudFormation Guard Rule
• **Control behavior:** Proactive
• **Resource types:** AWS::IAM::Policy, AWS::IAM::Role, AWS::IAM::User, AWS::IAM::Group
• **AWS CloudFormation guard rule:** [CT.IAM.PR.1 rule specification (p. 859)](p. 859)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.IAM.PR.1 rule specification (p. 859)](p. 859)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.IAM.PR.1 example templates (p. 864)](p. 864)

**Explanation**

IAM policies define a set of privileges that are granted to users, groups, or roles. In alignment with industry-standard security advice, AWS recommends that your policies grant least privilege, which means to grant only the permissions that are required to perform a task. When you provide full administrative privileges instead of the minimum set of permissions that the user requires, you may expose the resources to unwanted actions.

Instead of allowing full administrative privileges, determine the specific actions that your users must carry out, and then craft policies that let the users perform only those tasks. It is more secure to start with a minimum set of permissions and grant additional permissions when necessary. Do not start with lenient permissions and try to tighten them later.

Remove IAM policies that have a statement with Effect: Allow that permit Action: * over Resource: *.

**Usage considerations**

- This control applies only to IAM inline policies with statements that contain an Effect of Allow and that contain both the Action and the Resource element.

**Remediation for rule failure**

Remove IAM inline policy statements with Effect: Allow that permit Action: * over Resource: *.

The examples that follow show how to implement this remediation.

**IAM Policy - Example One**

IAM inline policy configured to allow retrieval of objects from an Amazon S3 bucket. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "IAMPolicy": {
        "Type": "AWS::IAM::Policy",
        "Properties": {
            "PolicyName": "sample-inline-policy",
            "Roles": [
                "Ref": "IAMRole"
            ]
        }
    }
}
```
“PolicyDocument”: {  
  “Version”: ”2012-10-17“,  
  “Statement”: [  
    {  
      “Effect”: ”Allow“,  
      “Action”: [  
        “s3:GetObject“  
      ],  
      “Resource”: [  
        “arn:aws:s3:::samplebucket/*“  
      ]  
    }  
  ]  
}

YAML example

IAMPolicy:  
Type: AWS::IAM::Policy  
Properties:  
  PolicyName: sample-inline-policy  
  Roles:  
    - !Ref 'IAMRole'  
  PolicyDocument:  
    Version: '2012-10-17'  
    Statement:  
      - Effect: Allow  
        Action:  
          - s3:GetObject  
        Resource:  
          - arn:aws:s3:::samplebucket/*

The examples that follow show how to implement this remediation.

IAM Role - Example Two

IAM role configured with an inline policy allowing retrieval of objects from an Amazon S3 bucket. The example is shown in JSON and in YAML.

JSON example

[  
  "IAMRole": {  
    "Type": "AWS::IAM::Role",  
    "Properties": {  
      "AssumeRolePolicyDocument": {  
        "Version": "2012-10-17",  
        "Statement": [  
          {  
            "Effect": "Allow",  
            "Principal": {  
              "AWS": {  
                "Ref": "AWS::AccountId"  
              }  
            },  
            "Action": [  
              "s3:GetObject"  
            ]  
          }  
        ]  
      }  
    }  
  }  
]
YAML example

IAMRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
      - Effect: Allow
        Principal:
          AWS: !Ref 'AWS::AccountId'
        Action:
          - sts:AssumeRole
      Policies:
      - PolicyName: sample-inline-policy
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
          - Effect: Allow
            Action:
              - s3:GetObject
            Resource:
              - arn:aws:s3:::samplebucket/*

CT.IAM.PR.1 rule specification

# ###################################
##       Rule Specification        ##
####################################
#
# Rule Name:
859
iam_inline_policy_no_statements_with_admin_access_check

# Description:
This control checks that AWS Identity and Access Management (IAM) inline policies do not include "Effect": "Allow" with "Action": "*" over "Resource": "*".

# Reports on:
AWS::IAM::Policy, AWS::IAM::Role, AWS::IAM::User, AWS::IAM::Group

# Evaluates:
AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
None

# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain any IAM policy, IAM role, IAM user or IAM group resources
# Then: SKIP
# Scenario: 2
# Given: The input document contains an IAM policy resource
# And: The policy has no statements with 'Effect' set to 'Allow'
# Then: SKIP
# Scenario: 3
# Given: The input document contains an IAM policy resource
# And: The policy has a statement with 'Effect' set to 'Allow'
# And: The policy does not have both Action and resource statements
# Then: SKIP
# Scenario: 4
# Given: The input document contains an IAM role, IAM user or IAM group resources
# And: 'Policies' is not provided or is an empty list
# Then: SKIP
# Scenario: 5
# Given: The input document contains an IAM role, IAM user or IAM group resources
# And: 'Policies' is provided as a non-empty list
# And: All IAM policy documents in 'Policies' have no statements with 'Effect' set to 'Allow'
# Then: SKIP
# Scenario: 6
# Given: The input document contains an IAM policy resource
# And: The policy has a statement with 'Effect' set to 'Allow'
# And: The policy statement has one or more Action statements and one or more Resource statements
# And: At least one Action statement allows all actions (Action value of '*')
# And: At least one Resource statement is a wildcard representing all resources (Resource value of '*')
# Then: FAIL
# Scenario: 7
# Given: The input document contains an IAM role, IAM user or IAM group resources
# And: 'Policies' is provided as a non-empty list
# And: IAM policy document in 'Policies' has a statement with 'Effect' set to 'Allow'
And: The policy has one or more 'Action' statements
And: At least one Action statement allows all actions (Action value of '*')
And: At least one Resource statement is a wildcard representing all resources (Resource value of '*')
Then: FAIL

Scenario: 8
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document contains an IAM policy resource
And: The policy has a statement with 'Effect' set to 'Allow'
And: The policy has one or more Action statements and one or more Resource statements
And: No Action statements allow administrator access (Action value of '*')
And: No Resources are wildcards representing all resources (Resource value of '*')
Then: PASS

Scenario: 9
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document contains an IAM role, IAM user or IAM group resources
And: 'Policies' is provided as a non-empty list
And: IAM policy document in 'Policies' has a statement with 'Effect' set to 'Allow'
And: The policy has one or more 'Action' statements
And: No Action statements allow administrator access (Action value of '*')
And: No Resources are wildcards representing all resources (Resource value of '*')
Then: PASS

# Constants
let AWS_IAM_POLICY_TYPE = "AWS::IAM::Policy"
let AWS_IAM_ROLE_TYPE = "AWS::IAM::Role"
let AWS_IAM_USER_TYPE = "AWS::IAM::User"
let AWS_IAM_GROUP_TYPE = "AWS::IAM::Group"
let INPUT_DOCUMENT = this

# Assignments
let iam_policies = Resources.*[ Type == %AWS_IAM_POLICY_TYPE ]
let iam_principals = Resources.*[
  Type == %AWS_IAM_ROLE_TYPE or
  Type == %AWS_IAM_USER_TYPE or
  Type == %AWS_IAM_GROUP_TYPE
]

# Primary Rules
rule iam_inline_policy_no_statements_with_admin_access_check when is_cfn_template(%INPUT_DOCUMENT) %iam_policies not empty {
    check_policy(%iam_policies.Properties)
    <<
    [CT.IAM.PR.1]: Require that an AWS Identity and Access Management (IAM) inline policy does not have a statement that includes "*" in the Action and Resource elements
    [FIX]: Remove IAM inline policy statements with "Effect": "Allow" that permit "Action": "*" over "Resource": "*".
    >>
}

rule iam_inline_policy_no_statements_with_admin_access_check when is_cfn_hook(%INPUT_DOCUMENT, %AWS_IAM_POLICY_TYPE) {
    check_policy(%INPUT_DOCUMENT.%AWS_IAM_POLICY_TYPE.resourceProperties)
    <<
    [CT.IAM.PR.1]: Require that an AWS Identity and Access Management (IAM) inline policy does not have a statement that includes "*" in the Action and Resource elements
    [FIX]: Remove IAM inline policy statements with "Effect": "Allow" that permit "Action": "*" over "Resource": "*".
    >>
}
[FIX]: Remove IAM inline policy statements with "Effect": "Allow" that permit
"Action": "*" over "Resource": "*".

rule iam_inline_policy_no_statements_with_admin_access_check when
  is_cfn_template(%INPUT_DOCUMENT)
%iam_principals not empty
{
  check_principal(%iam_principals.Properties)
<<
[CT.IAM.PR.1]: Require that an AWS Identity and Access Management (IAM) inline
policy does not have a statement that includes "*" in the Action and Resource elements
[FIX]: Remove IAM inline policy statements with "Effect": "Allow" that permit
"Action": "*" over "Resource": "*".
>>
}

rule check_policy(policy) {
  %policy[
    filter_policy_document_with_statement_provided(this)
  ]{
    PolicyDocument {
      check_statement(Statement)
    }
  }
}

rule check_principal(iam_principal) {
  %iam_principal [
filter_iam_principal_with_inline_policy_provided(this)
] {
    Policies[*] {
        check_policy(this)
    }
}

rule check_statement(statement) {
    %statement {
        filter_allow_on_action_and_resource(this)
    } {
        Action exists
        Resource exists
        check_admin_access(Action, Resource)
    }
}

rule filter_allow_on_action_and_resource(statement) {
    %statement {
        Effect == "Allow"
        Action exists
        Resource exists
    }
}

rule filter_policy_document_with_statement_provided(policy) {
    %policy {
        PolicyDocument exists
        PolicyDocument is_struct
        PolicyDocument {
            Statement exists
            filter_statement_non_empty_list(Statement) or
            Statement is_struct
        }
    }
}

rule filter_iam_principal_with_inline_policy_provided(iam_principal) {
    %iam_principal {
        Policies exists
        Policies is_list
        Policies not empty
    }
}

rule filter_statement_non_empty_list(statement) {
    %statement {
        this is_list
        this not empty
    }
}

rule check_admin_access(actions, resources) {
    when some %actions[*] == "*" {
        %resources[*] != "*"
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
CT.IAM.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  IAMRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              AWS:
                Ref: AWS::AccountId
            Action:
              - sts:AssumeRole
      IAMPolicy:
        Type: AWS::IAM::Policy
        Properties:
          PolicyName:
           Fn::Sub: ${AWS::StackName}-inline-policy
        Roles:
          - Ref: IAMRole
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - s3:GetObject
              Resource:
                - arn:aws:s3:::examplebucket/*
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```yaml
Resources:
  IAMRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              AWS:
                Ref: AWS::AccountId
            Action:
              - sts:AssumeRole
      IAMPolicy:
      Type: AWS::IAM::Policy
      Properties:
        PolicyName:
          Fn::Sub: ${AWS::StackName}-inline-policy
        Roles:
          - Ref: IAMRole
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - s3:GetObject
              Resource:
                - arn:aws:s3:::examplebucket/*
```
[CT.IAM.PR.2] Require that AWS Identity and Access Management (IAM) customer-managed policies do not contain a statement that includes "*" in the Action and Resource elements

This control checks whether AWS Identity and Access Management (IAM) customer managed policies do not include Effect: Allow with Action: * over Resource: *.

- **Control objective**: Enforce least privilege
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::IAM::ManagedPolicy
- **AWS CloudFormation guard rule**: [CT.IAM.PR.2 rule specification (p. 867)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.IAM.PR.2 rule specification (p. 867)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.IAM.PR.2 example templates (p. 869)]

**Explanation**

IAM policies define a set of privileges that are granted to users, groups, or roles. In alignment with industry-standard security advice, AWS recommends that your policies grant least privilege, which means to grant only the permissions that are required to perform a task. When you provide full privileges instead of the minimum set of permissions that the user requires, you may expose the resources to unwanted actions.

Instead of allowing full privileges, determine the specific actions that your users must carry out, and then craft policies that let the users perform only those tasks. It is more secure to start with a minimum set of permissions and grant additional permissions when necessary. Do not start with lenient permissions and try to tighten them later.

Remove IAM policies that have a statement with Effect: Allow that permit Action: * over Resource: *.

**Usage considerations**

- This control checks IAM customer-managed policies only. It does not check inline and AWS-managed policies.
- This control applies only to IAM inline policies with statements that contain an Effect of Allow and that contain both the Action and the Resource element.
Remediation for rule failure

Remove IAM inline policy statements with Effect: Allow that permit Action: * over Resource: *.

The examples that follow show how to implement this remediation.

IAM Managed Policy - Example

IAM managed policy configured to allow retrieval of objects from an Amazon S3 bucket. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "IAMManagedPolicy": {
        "Type": "AWS::IAM::ManagedPolicy",
        "Properties": {
            "Roles": [
                {
                    "Ref": "IAMRole"
                }
            ],
            "PolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": ["s3:GetObject"],
                        "Resource": ["arn:aws:s3:::samplebucket/*"]
                    }
                ]
            }
        }
    }
}
```

**YAML example**

```yaml
IAMManagedPolicy:
  Type: AWS::IAM::ManagedPolicy
  Properties:
    Roles:
      - !Ref 'IAMRole'
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Action:
            - s3:GetObject
          Resource:
            - arn:aws:s3:::samplebucket/*
```
CT.IAM.PR.2 rule specification

# Rule Name:
# iam.managed_policy_no_statements_with_admin_access_check

# Description:
# This control checks whether AWS Identity and Access Management (IAM) customer managed policies do not include "Effect": "Allow" with "Action": "*" over "Resource": "*".

# Reports on:
# AWS::IAM::ManagedPolicy

# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain any IAM managed policy resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an IAM managed policy resource
# And: The policy has no statements with 'Effect' set to 'Allow'
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an IAM managed policy resource
# And: The policy has a statement with 'Effect' set to 'Allow'
# And: The policy does not have both Action and Resource statements
# Then: SKIP
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an IAM managed policy resource
# And: The policy has a statement with 'Effect' set to 'Allow'
# And: The policy statement has one or more Action statements and one or more Resource statements
# And: Within a single policy statement at least one Action statement allows all actions (Action value of '*')
# And: Within the same policy statement at least one Resource statement is a wildcard representing all resources (Resource value of '*')
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an IAM managed policy resource
# And: The policy has a statement with 'Effect' set to 'Allow'
# And: The policy has one or more Action statements and one or more Resource statements
# And: Within a single policy statement no Action statements allow all actions (Action value of '*')
# And: Within the same policy statement no Resources are wildcards representing all
resources (Resource value of '**) Then: PASS

# Constants
let AWS_IAM_MANAGED_POLICY_TYPE = "AWS::IAM::ManagedPolicy"
let INPUT_DOCUMENT = this

# Assignments
let iam_managed_policies = Resources.*[ Type == %AWS_IAM_MANAGED_POLICY_TYPE ]

# Primary Rules

rule iam_managed_policy_no_statements_with_admin_access_check when
is_cfn_template(%INPUT_DOCUMENT)
%iam_managed_policies not empty {
    check(%iam_managed_policies.Properties)
    <<
    [CT.IAM.PR.2]: Require that AWS Identity and Access Management (IAM) customer-
    managed policies do not contain a statement that includes "**" in the Action and Resource
    elements
    [FIX]: Remove IAM inline policy statements with "Effect": "Allow" that permit
    "Action": "**" over "Resource": "**".
    >>
}

rule iam_managed_policy_no_statements_with_admin_access_check when
is_cfn_hook(%INPUT_DOCUMENT, %AWS_IAM_MANAGED_POLICY_TYPE) {
    check(%INPUT_DOCUMENT.%AWS_IAM_MANAGED_POLICY_TYPE.resourceProperties)
    <<
    [CT.IAM.PR.2]: Require that AWS Identity and Access Management (IAM) customer-
    managed policies do not contain a statement that includes "**" in the Action and Resource
    elements
    [FIX]: Remove IAM inline policy statements with "Effect": "Allow" that permit
    "Action": "**" over "Resource": "**".
    >>
}

# Parameterized Rules

rule check(policy) {
    %policy [
        filter_policy_document_with_statement_provided(this)
    ] {
        PolicyDocument {
            check_statement(Statement)
        }
    }
}

rule check_statement(statement) {
    %statement [
        filter_allow_on_action_and_resource(this)
    ] {
        Action exists
        Resource exists
        check_admin_access(Action, Resource)
    }
}
rule filter_allow_on_action_and_resource(statement) {
    %statement {
        Effect == "Allow"
        Action exists
        Resource exists
    }
}

rule filter_policy_document_with_statement_provided(policy) {
    %policy {
        PolicyDocument exists
        PolicyDocument is_struct
        PolicyDocument {
            Statement exists
            filter_statement_non_empty_list(Statement) or
            Statement is_struct
        }
    }
}

rule filter_statement_non_empty_list(statement) {
    %statement {
        this is_list
        this not empty
    }
}

rule check_admin_access(actions, resources) {
    when some %actions[*] == "*
    %resources[*] != "*" {
    }
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

---

**CT.IAM.PR.2 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

Resources:
IAMRole:
    Type: AWS::IAM::Role
Properties:
    AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
            - Effect: Allow
[CT.IAM.PR.3] Require that AWS Identity and Access Management (IAM) customer-managed policies do not have wildcard service actions

This control checks that AWS Identity and Access Management (IAM) customer-managed policies do not contain statements of Effect: Allow with Action: Service:* (for example, s3:*') for individual AWS services, and that the policies do not use the combination of NotAction with an Effect of Allow.

- **Control objective:** Enforce least privilege
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
• **Resource types:** AWS::IAM::ManagedPolicy

• **AWS CloudFormation guard rule:** [CT.IAM.PR.3 rule specification (p. 872)](#)

## Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.IAM.PR.3 rule specification (p. 872)](#)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.IAM.PR.3 example templates (p. 875)](#)

## Explanation

When you assign permissions to AWS services, it is important to scope the allowed IAM actions in your IAM policies. We recommend that you provision least-privilege permissions by restricting IAM policies to required actions only. Overly permissive policies can lead to privilege escalation, if the policies are attached to an IAM principal that may not require the permission.

### Usage considerations

- This control checks IAM customer-managed policies only. It does not check inline and AWS-managed policies.
- This control applies only to IAM customer-managed policies with statements that contain an `Effect` of `Allow` with an `Action` or `NotAction` element present.

### Remediation for rule failure

Remove statements from IAM customer-managed policies with `Effect`: `Allow` and `Action`: `service:*` or `Effect`: `Allow` and `NotAction`.

The examples that follow show how to implement this remediation.

### IAM Managed Policy - Example

IAM managed policy configured to allow the Amazon S3 `ListBucket` action. The example is shown in JSON and in YAML.

#### JSON example

```json
{
    "IAMManagedPolicy": {
        "Type": "AWS::IAM::ManagedPolicy",
        "Properties": {
            "Roles": [
                {
                    "Ref": "IAMRole"
                }
            ],
            "PolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Action": [
                            "s3:ListBucket"
                        ],
                        "Resource": ["*"
                    }
                ]
            }
        }
    }
}
```
YAML example

IAMManagedPolicy:
  Type: AWS::IAM::ManagedPolicy
  Properties:
    Roles:
      - !Ref 'IAMRole'
  PolicyDocument:
    Version: '2012-10-17'
    Statement:
      - Effect: Allow
        Action:
          - s3:ListBucket
        Resource:
          - '*'

CT.IAM.PR.3 rule specification

# ###################################################################
#       Rule Specification       #
# ###################################################################
#
# Rule Name:
#   iam_managed_policy_no_statements_with_full_access_check
#
# Description:
#   This control checks that AWS Identity and Access Management (IAM) customer-managed policies do not contain statements of "Effect": "Allow" with "Action": "Service:*" (for example, s3:*) for individual AWS services, and that the policies do not use the combination of "NotAction" with an "Effect" of "Allow".
#
# Reports on:
#   AWS::IAM::ManagedPolicy
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document does not contain any IAM Managed Policy resources
#       Then: SKIP
#
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document contains an IAM Managed Policy resource
#       And: The policy has no statements with 'Effect' set to 'Allow'
#       Then: SKIP
#   Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an IAM Managed Policy resource
And: The policy has a statement with 'Effect' set to 'Allow'
And: The policy has one or more 'Action' statements
And: At least one 'Action' statement allows full access to a service ('Action' has a value 'service:*')
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an IAM Managed Policy resource
And: The policy has a statement with 'Effect' set to 'Allow'
And: The policy has one or more 'NotAction' statements
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an IAM Managed Policy resource
And: The policy has a statement with 'Effect' set to 'Allow'
And: The policy has one or more 'Action' statements
And: No 'Action' statements allow full access to a service ('Action' does not have a value 'service:*')
Then: PASS

# Constants

let AWS_IAM_MANAGED_POLICY_TYPE = "AWS::IAM::ManagedPolicy"
let WILDCARD_ACTION_PATTERN = /^\[\w\]*\[:]*\*$/
let INPUT_DOCUMENT = this

# Assignments

let iam_managed_policies = Resources.*[ Type == %AWS_IAM_MANAGED_POLICY_TYPE ]

# Primary Rules

rule iam_managed_policy_no_statements_with_full_access_check when
is_cfn_template(%INPUT_DOCUMENT)
%iam_managed_policies not empty {
    check(%iam_managed_policies.Properties)
    <<
    [CT.IAM.PR.3]: Require that AWS Identity and Access Management (IAM) customer-managed policies do not have wildcard service actions
    [FIX]: Remove statements from IAM customer-managed policies with "Effect": "Allow" and "Action": "service:*" or "Effect": "Allow" and "NotAction".
    >>
}

rule iam_managed_policy_no_statements_with_full_access_check when
is_cfn_hook(%INPUT_DOCUMENT, %AWS_IAM_MANAGED_POLICY_TYPE) {
    check(%INPUT_DOCUMENT.%AWS_IAM_MANAGED_POLICY_TYPE.resourceProperties)
    <<
    [CT.IAM.PR.3]: Require that AWS Identity and Access Management (IAM) customer-managed policies do not have wildcard service actions
    [FIX]: Remove statements from IAM customer-managed policies with "Effect": "Allow" and "Action": "service:*" or "Effect": "Allow" and "NotAction".
    >>
}

# Parameterized Rules
# rule check(policy) {
  %policy [%policy_document_with_statement_provided(this)
    %PolicyDocument [
      check_statement_no_wildcard_actions(Statement)
      check_statement_no_not_action(Statement)
    ]
  ]
}

rule check_statement_no_wildcard_actions(statement) {
  %statement [%filter_allow_on_action(this)
    % Action exists
    check_no_wildcard_action(Action)
  ]
}

rule check_statement_no_not_action(statement) {
  %statement [%filter_allow(this)
    % NotAction not exists
  ]
}

rule filter_allow_on_action(statement) {
  %statement {
    Effect == "Allow"
    Action exists
  }
}

rule filter_allow(statement) {
  %statement {
    Effect == "Allow"
  }
}

rule filter_policy_document_with_statement_provided(policy) {
  %policy {
    PolicyDocument exists
    PolicyDocument is_struct
    PolicyDocument {
      Statement exists
      filter_statement_non_empty_list(Statement) or
      Statement is_struct
    }
  }
}

rule filter_statement_non_empty_list(statement) {
  %statement {
    this is_list
    this not empty
  }
}

rule check_no_wildcard_action(actions) {
  %actions[*] {
    this != %WILDCARD_ACTION_PATTERN
  }
}
# Utility Rules

```bash
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
```

```bash
rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

## CT.IAM.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

### PASS Example

- **Use this template to verify a compliant resource creation.**

```bash
Resources:
IAMRole:
    Type: AWS::IAM::Role
    Properties:
        AssumeRolePolicyDocument:
            Version: '2012-10-17'
            Statement:
                - Effect: Allow
                  Principal:
                    AWS:
                        Ref: AWS::AccountId
                  Action:
                    - sts:AssumeRole
IAMManagedPolicy:
    Type: AWS::IAM::ManagedPolicy
    Properties:
        Roles:
            - Ref: IAMRole
    PolicyDocument:
            Version: '2012-10-17'
            Statement:
                - Effect: Allow
                  Action:
                    - s3:ListBucket
                  Resource:
                    - '*'
```

### FAIL Example

- **Use this template to verify that the control prevents non-compliant resource creation.**

```bash
Resources:
IAMRole:
    Type: AWS::IAM::Role
    Properties:
        AssumeRolePolicyDocument:
            Version: '2012-10-17'
            Statement:
```

875
[CT.IAM.PR.4] Require that an AWS Identity and Access Management (IAM) user does not have an inline or managed policy attached

This control checks whether your AWS Identity and Access Management (IAM) user has inline or managed (AWS and customer) policies directly attached. Instead, IAM users should inherit permissions from IAM groups or roles.

- **Control objective**: Enforce least privilege
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::IAM::User, AWS::IAM::Policy, AWS::IAM::ManagedPolicy
- **AWS CloudFormation guard rule**: [CT.IAM.PR.4 rule specification](p. 878)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.IAM.PR.4 rule specification](p. 878)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.IAM.PR.4 example templates](p. 881)

Explanation

By default, IAM users, groups, and roles have no access to AWS resources. IAM policies grant privileges to users, groups, or roles. We recommend that you apply IAM policies directly to groups and roles, but not to users. As the number of users grows, assigning privileges at the group or role level reduces the complexity of access management. Reducing access management complexity may in turn reduce the opportunity for a principal to receive or retain excessive privileges inadvertently.

Remediation for rule failure

Configure IAM users to inherit permissions from IAM groups.

The examples that follow show how to implement this remediation.

**IAM User - Example**

IAM user configured with no IAM policy or managed policy attachments. The example is shown in JSON and in YAML.
Proactive controls

JSON example

```json
{
   "IAMUser": {
      "Type": "AWS::IAM::User"
   }
}
```

YAML example

```yaml
IAMUser:
    Type: AWS::IAM::User
```

The examples that follow show how to implement this remediation.

**IAM Policy - Example**

IAM policy configured with no IAM user associations. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "IAMPolicy": {
      "Type": "AWS::IAM::Policy",
      "Properties": {
         "PolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
               {
                  "Effect": "Allow",
                  "Action": [
                     "cloudformation:DescribeStacks"
                  ],
                  "Resource": "*"
               }
            ]
         },
         "PolicyName": "sample-policy",
         "Roles": [
            {
               "Ref": "IAMRole"
            }
         ]
      }
   }
}
```

**YAML example**

```yaml
IAMPolicy:
    Type: AWS::IAM::Policy
    Properties:
       PolicyDocument:
```
The examples that follow show how to implement this remediation.

**IAM Managed Policy - Example**

IAM managed policy configured with no IAM user associations. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "IAMManagedPolicy": {
    "Type": "AWS::IAM::ManagedPolicy",
    "Properties": {
      "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": [
              "cloudformation:DescribeStacks"
            ],
            "Resource": "*"
          }
        ]
      }
    }
  }
}
```

**YAML example**

```yaml
IAMManagedPolicy:
  Type: AWS::IAM::ManagedPolicy
  Properties:
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Action:
            - cloudformation:DescribeStacks
          Resource: '*'
```

**CT.IAM.PR.4 rule specification**
# Rule Specification

# Rule Identifier:
# iam_user_no_policies_check

# Description:
# This control checks whether your AWS Identity and Access Management (IAM) user has
# inline or managed (AWS and customer) policies directly attached. Instead, IAM users should
# inherit permissions from IAM groups or roles.

# Reports on:
# AWS::IAM::User, AWS::IAM::Policy, AWS::IAM::ManagedPolicy

# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#       And: The input document does not contain any IAM user, policy or managed policy
#             resources
#       Then: SKIP
#   Scenario: 2
#   Scenario: 3
#   Scenario: 4
#   Scenario: 5

# Constants

let IAM_USER_TYPE = "AWS::IAM::User"
let IAM_POLICY_TYPE = "AWS::IAM::Policy"
let IAM_MANAGED_POLICY_TYPE = "AWS::IAM::ManagedPolicy"
let INPUT_DOCUMENT = this

# Assignments

let iam_users = Resources.*[ Type == %IAM_USER_TYPE ]
let iam_policies = Resources.*(Type == %IAM_POLICY_TYPE)
let iam_managed_policies = Resources.*(Type == %IAM_MANAGED_POLICY_TYPE)

# Primary Rules
#
rule iam_user_no_policies_check when is_cfn_template(%INPUT_DOCUMENT)
  %iam_users not empty {
    check_user(%iam_users.Properties)
    <<
    [CT.IAM.PR.4]: Require that an AWS Identity and Access Management (IAM) user does not have an inline or managed policy attached
    [FIX]: Configure IAM users to inherit permissions from IAM groups.
    >>
  }

rule iam_user_no_policies_check when is_cfn_template(%INPUT_DOCUMENT)
  %iam_policies not empty {
    check_policy(%iam_policies.Properties)
    <<
    [CT.IAM.PR.4]: Require that an AWS Identity and Access Management (IAM) user does not have an inline or managed policy attached
    [FIX]: Configure IAM users to inherit permissions from IAM groups.
    >>
  }

rule iam_user_no_policies_check when is_cfn_template(%INPUT_DOCUMENT)
  %iam_managed_policies not empty {
    check_policy(%iam_managed_policies.Properties)
    <<
    [CT.IAM.PR.4]: Require that an AWS Identity and Access Management (IAM) user does not have an inline or managed policy attached
    [FIX]: Configure IAM users to inherit permissions from IAM groups.
    >>
  }

rule iam_user_no_policies_check when is_cfn_hook(%INPUT_DOCUMENT, %IAM_USER_TYPE) {
  check_user(%INPUT_DOCUMENT.%IAM_USER_TYPE.resourceProperties)
  <<
  [CT.IAM.PR.4]: Require that an AWS Identity and Access Management (IAM) user does not have an inline or managed policy attached
  [FIX]: Configure IAM users to inherit permissions from IAM groups.
  >>
}

rule iam_user_no_policies_check when is_cfn_hook(%INPUT_DOCUMENT, %IAM_POLICY_TYPE) {
  check_policy(%INPUT_DOCUMENT.%IAM_POLICY_TYPE.resourceProperties)
  <<
  [CT.IAM.PR.4]: Require that an AWS Identity and Access Management (IAM) user does not have an inline or managed policy attached
  [FIX]: Configure IAM users to inherit permissions from IAM groups.
  >>
}

rule iam_user_no_policies_check when is_cfn_hook(%INPUT_DOCUMENT, %IAM_MANAGED_POLICY_TYPE) {
  check_policy(%INPUT_DOCUMENT.%IAM_MANAGED_POLICY_TYPE.resourceProperties)
  <<
  [CT.IAM.PR.4]: Require that an AWS Identity and Access Management (IAM) user does not have an inline or managed policy attached
  [FIX]: Configure IAM users to inherit permissions from IAM groups.
  >>
}

# Parameterized Rules
# Proactive controls

```yaml
rule check_user(iam_user) {
  %iam_user {
    # Scenario 2 and 4
    Policies not exists or
    exists_and_is_empty_list(Policies)
    ManagedPolicyArns not exists or
    exists_and_is_empty_list(ManagedPolicyArns)
  }
}

rule check_policy(policy) {
  %policy {
    # Scenario 3 and 4
    Users not exists or
    exists_and_is_empty_list(Users)
  }
}

rule exists_and_is_empty_list(list_property) {
  %list_property {
    this is_list
    this empty
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists  or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

## CT.IAM.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

Resources:
- IAMUser:
  - Type: AWS::IAM::User
  - Properties: {}

**FAIL Example** - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
- IAMUser:
  - Type: AWS::IAM::User
Properties:
   ManagedPolicyArns:
   - arn:aws:iam::aws:policy/AdministratorAccess

[CT.IAM.PR.5] Requirement that AWS Identity and Access Management (IAM) inline policies do not have wildcard service actions

This control checks whether AWS Identity and Access Management (IAM) inline policies do not include Effect: Allow with Action: Service:* (e.g. s3:*). It ensures that IAM policies are configured to restrict service actions.

- **Control objective:** Enforce least privilege
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::IAM::Policy, AWS::IAM::Role, AWS::IAM::User, AWS::IAM::Group
- **AWS CloudFormation guard rule:** [CT.IAM.PR.5 rule specification (p. 886)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.IAM.PR.5 rule specification (p. 886)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.IAM.PR.5 example templates (p. 891)]

**Explanation**

When you assign permissions to AWS services, it is important to scope the allowed IAM actions in your IAM policies. You should restrict IAM actions to only those actions that are needed. This helps you to provision least privilege permissions. Overly permissive policies might lead to privilege escalation if the policies are attached to an IAM principal that might not require the permission.

**Usage considerations**

- This control only applies to IAM inline policies with statements that contain an Effect of Allow with an Action or NotAction element present.
- This control only applies to IAM role, user or group resources with one or more inline policies and IAM policy resources with one or more statements configured.

**Remediation for rule failure**

Remove statements from IAM inline policies with Effect: Allow and Action: service:* or Effect: Allow and NotAction.

The examples that follow show how to implement this remediation.

**IAM Inline Policy - Example One**

IAM role configured with an inline policy allowing the S3 ListBucket action. The example is shown in JSON and in YAML.

**JSON example**

---

882
IAMRole:
Type: AWS::IAM::Role
Properties:
  AssumeRolePolicyDocument:
    Version: '2012-10-17'
    Statement:
      - Effect: Allow
        Principal:
          AWS: !Ref 'AWS::AccountId'
        Action:
          - sts:AssumeRole
  Policies:
    - PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action:
              - s3:ListBucket
            Resource:
              - '*'
        PolicyName: "sample-policy"
Proactive controls

**Action:**
- s3:ListBucket

**Resource:**
- `'*'`

**PolicyName:** sample-policy

The examples that follow show how to implement this remediation.

**IAM Inline Policy - Example Two**

IAM user configured with an inline policy allowing the S3 ListBucket action. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "IAMUser": {
    "Type": "AWS::IAM::User",
    "Properties": {
      "Policies": [
        {
          "PolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
              {
                "Effect": "Allow",
                "Action": ["s3:ListBucket"],
                "Resource": ["'*'"]
              }
            ]
          },
          "PolicyName": "sample-policy"
        }
      ]
    }
  }
}
```

**YAML example**

```
IAMUser:
  Type: AWS::IAM::User
  Properties:
    Policies:
      - PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - s3:ListBucket
              Resource:
                - "'*'"
          PolicyName: sample-policy
```
The examples that follow show how to implement this remediation.

**IAM Inline Policy - Example Three**

IAM group configured with an inline policy allowing the S3 ListBucket action. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "IAMGroup": {
        "Type": "AWS::IAM::Group",
        "Properties": {
            "Policies": [
                {
                    "PolicyDocument": {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Action": ["s3:ListBucket"],
                                "Resource": ["*"]
                            }
                        ]
                    },
                    "PolicyName": "sample-policy"
                }
            ]
        }
    }
}
```

**YAML example**

```yaml
IAMGroup:
  Type: AWS::IAM::Group
  Properties:
    Policies:
      - PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action: ["s3:ListBucket"],
              Resource: ["*"]
        PolicyName: sample-policy
```

The examples that follow show how to implement this remediation.
IAM Inline Policy - Example Four

IAM policy associated with an IAM role as an inline policy and configured to allow the S3 ListBucket action. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "IAMPolicy": {
      "Type": "AWS::IAM::Policy",
      "Properties": {
         "PolicyName": "sample-policy",
         "Roles": [
            {
               "Ref": "IAMRole"
            }
         ],
         "PolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
               {
                  "Effect": "Allow",
                  "Action": ["s3:ListBucket"],
                  "Resource": ["*"]
               }
            ]
         }
      }
   }
}
```

**YAML example**

```yaml
IAMPolicy:
  Type: AWS::IAM::Policy
  Properties:
    PolicyName: sample-policy
    Roles:
    - !Ref 'IAMRole'
  PolicyDocument:
    Version: '2012-10-17'
    Statement:
    - Effect: Allow
      Action:
      - s3:ListBucket
      Resource:
      - '*'
```

**CT.IAM.PR.5 rule specification**

```
# ##################################################################
## Rule Specification    ##
```
# Rule Name:
# iam_inline_policy_no_statements_with_full_access_check
#
# Description:
# This control checks whether AWS Identity and Access Management (IAM) inline policies
do not include "Effect": "Allow" with "Action": "Service:*" (e.g. s3:*) for individual AWS
services or use the combination of "NotAction" with an "Effect" of "Allow".
#
# Reports on:
# AWS::IAM::Policy, AWS::IAM::Role, AWS::IAM::User, AWS::IAM::Group
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document does not contain any IAM policy, IAM role, IAM user or IAM
group resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an IAM policy resource
# And: The policy has no statements with 'Effect' set to 'Allow'
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an IAM role, IAM user or IAM group resource
# And: 'Policies' is not provided or is an empty list
# Then: SKIP
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an IAM role, IAM user or IAM group resource
# And: 'Policies' is provided as a non-empty list
# And: All IAM policy documents in 'Policies' have no statements with 'Effect' set to
'Allow'
# Then: SKIP
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an IAM policy resource
# And: The policy has a statement with 'Effect' set to 'Allow'
# And: The policy has one or more 'Action' statements
# And: 'Action' statement allows full access to a service ('Action' has a value
'service:*')
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an IAM policy resource
# And: The policy has a statement with 'Effect' set to 'Allow'
# And: The policy has one or more 'NotAction' statements
# Then: FAIL
# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an IAM role, IAM user or IAM group resource
# And: 'Policies' is provided as a non-empty list
# And: IAM policy document in 'Policies' has a statement with 'Effect' set to 'Allow'
# And: The policy has one or more 'Action' statements
And: 'Action' statement allows full access to a service ('Action' has a value 'service:*')
Then: FAIL

Scenario: 8
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an IAM role, IAM user or IAM user resource
And: 'Policies' is provided as a non-empty list
And: At least one IAM policy document in 'Policies' has a statement with 'Effect' set to 'Allow'
And: The policy has one or more 'NotAction' statements
Then: FAIL

Scenario: 9
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an IAM policy resource
And: The policy has a statement with 'Effect' set to 'Allow'
And: The policy has one or more 'Action' statements
And: No 'Action' statements allow full access to a service ('Action' does not have a value 'service:*')
Then: PASS

Scenario: 10
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an IAM role, IAM user or IAM user resource
And: 'Policies' is provided as a non-empty list
And: At least one IAM policy document in 'Policies' has a statement with 'Effect' set to 'Allow'
And: The policy has one or more 'Action' statements
And: No 'Action' statements allow full access to a service ('Action' does not have a value 'service:*')
Then: PASS

# Constants

let AWS_IAM_POLICY_TYPE = "AWS::IAM::Policy"
let AWS_IAM_ROLE_TYPE = "AWS::IAM::Role"
let AWS_IAM_USER_TYPE = "AWS::IAM::User"
let AWS_IAM_GROUP_TYPE = "AWS::IAM::Group"
let WILDCARD_ACTION_PATTERN = /^\[\w\]*\[:\]*\*$/

let INPUT_DOCUMENT = this

# Assignments

let iam_policies = Resources.*[ Type == %AWS_IAM_POLICY_TYPE ]
let iam_principals = Resources.*[
  Type == %AWS_IAM_ROLE_TYPE or
  Type == %AWS_IAM_USER_TYPE or
  Type == %AWS_IAM_GROUP_TYPE
]

# Primary Rules

rule iam_inline_policy_no_statements_with_full_access_check when
  is_cfn_template(%INPUT_DOCUMENT)
  %iam_policies not empty {
    check_policy(%iam_policies.Properties)
    <<
      [CT.IAM.PR.5]: Require that AWS Identity and Access Management (IAM) inline policies do not have wildcard service actions
      [FIX]: Remove statements from IAM inline policies with "Effect": "Allow" and "Action": "service:*" or "Effect": "Allow" and "NotAction".
    >>
  }
rule iam_inline_policy_no_statements_with_full_access_check when
is_cfn_hook(%INPUT_DOCUMENT, %AWS_IAM_POLICY_TYPE) {
    check_policy(%INPUT_DOCUMENT.%AWS_IAM_POLICY_TYPE.resourceProperties)
    <<
        [CT.IAM.PR.5]: Require that AWS Identity and Access Management (IAM) inline policies do not have wildcard service actions
        [FIX]: Remove statements from IAM inline policies with "Effect": "Allow" and "Action": "service:*" or "Effect": "Allow" and "NotAction".
        >>
}

rule iam_inline_policy_no_statements_with_full_access_check when
is_cfn_template(%INPUT_DOCUMENT) %iam_principals not empty
{
    check_principal(%iam_principals.Properties)
    <<
        [CT.IAM.PR.5]: Require that AWS Identity and Access Management (IAM) inline policies do not have wildcard service actions
        [FIX]: Remove statements from IAM inline policies with "Effect": "Allow" and "Action": "service:*" or "Effect": "Allow" and "NotAction".
        >>
}

rule iam_inline_policy_no_statements_with_full_access_check when
is_cfn_hook(%INPUT_DOCUMENT, %AWS_IAM_ROLE_TYPE) {
    check_principal(%INPUT_DOCUMENT.%AWS_IAM_ROLE_TYPE.resourceProperties)
    <<
        [CT.IAM.PR.5]: Require that AWS Identity and Access Management (IAM) inline policies do not have wildcard service actions
        [FIX]: Remove statements from IAM inline policies with "Effect": "Allow" and "Action": "service:*" or "Effect": "Allow" and "NotAction".
        >>
}

rule iam_inline_policy_no_statements_with_full_access_check when
is_cfn_hook(%INPUT_DOCUMENT, %AWS_IAM_USER_TYPE) {
    check_principal(%INPUT_DOCUMENT.%AWS_IAM_USER_TYPE.resourceProperties)
    <<
        [CT.IAM.PR.5]: Require that AWS Identity and Access Management (IAM) inline policies do not have wildcard service actions
        [FIX]: Remove statements from IAM inline policies with "Effect": "Allow" and "Action": "service:*" or "Effect": "Allow" and "NotAction".
        >>
}

rule iam_inline_policy_no_statements_with_full_access_check when
is_cfn_hook(%INPUT_DOCUMENT, %AWS_IAM_GROUP_TYPE) {
    check_principal(%INPUT_DOCUMENT.%AWS_IAM_GROUP_TYPE.resourceProperties)
    <<
        [CT.IAM.PR.5]: Require that AWS Identity and Access Management (IAM) inline policies do not have wildcard service actions
        [FIX]: Remove statements from IAM inline policies with "Effect": "Allow" and "Action": "service:*" or "Effect": "Allow" and "NotAction".
        >>
}

# Parameterized Rules
#
rule check_policy(policy) {
    %policy [
        filter_policy_document_with_statement_provided(this)
    ]
}
AWS Control Tower User Guide
Proactive controls

} {
  PolicyDocument {
    check_statement_no_wildcard_actions(Statement)
    check_statement_no_not_action(Statement)
  }
}

rule check_principal(iam_principal) {
  %iam_principal [
    filter_iam_principal_with_inline_policy_provided(this)
  ] {
    Policies[*] {
      check_policy(this)
    }
  }
}

rule check_statement_no_wildcard_actions(statement) {
  %statement [
    filter_allow_on_action(this)
  ] {
    Action exists
    check_no_wildcard_action(Action)
  }
}

rule check_statement_no_not_action(statement) {
  %statement [
    filter_allow(this)
  ] {
    NotAction not exists
  }
}

rule filter_allow_on_action(statement) {
  %statement {
    Effect == "Allow"
    Action exists
  }
}

rule filter_allow(statement) {
  %statement {
    Effect == "Allow"
  }
}

rule filter_policy_document_with_statement_provided(policy) {
  %policy {
    PolicyDocument exists
    PolicyDocument is_struct
    PolicyDocument {
      Statement exists
      filter_statement_non_empty_list(Statement) or
      Statement is_struct
    }
  }
}

rule filter_iam_principal_with_inline_policy_provided(iam_principal) {
  %iam_principal {
    Policies exists
    Policies is_list
    Policies not empty
  }
}


```asciidoc
# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

CT.IAM.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
IAMRole:
    Type: AWS::IAM::Role
    Properties:
        AssumeRolePolicyDocument:
            Version: '2012-10-17'
            Statement:
                - Effect: Allow
                    Principal:
                        AWS:
                            Ref: AWS::AccountId
                    Action:
                        - sts:AssumeRole
IAMPolicy:
    Type: AWS::IAM::Policy
    Properties:
        PolicyName:
            Fn::Sub: ${AWS::StackName}-example
        Roles:
            - Ref: IAMRole
        PolicyDocument:
            Version: '2012-10-17'
            Statement:
                - Effect: Allow
                    Action:
                        - s3:ListBucket
```

891
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
IAMRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            AWS:
              Ref: AWS::AccountId
          Action:
            - sts:AssumeRole
    IAMPolicy:
      Type: AWS::IAM::Policy
      Properties:
        PolicyName:
          Fn::Sub: ${AWS::StackName}-example
        Roles:
          - Ref: IAMRole
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action: s3:*
              Resource: '*'

AWS Key Management Service (AWS KMS) controls

Topics
- [CT.KMS.PR.1] Require any AWS KMS key to have rotation configured (p. 892)
- [CT.KMS.PR.2] Require that an AWS Key Management Service asymmetric key with RSA key material used for encryption has a key length greater than 2048 bits (p. 897)
- [CT.KMS.PR.3] Require an AWS Key Management Service key policy to have a statement that limits creation of AWS KMS grants to AWS services (p. 901)

[CT.KMS.PR.1] Require any AWS KMS key to have rotation configured

This control checks whether key rotation is enabled for AWS KMS customer managed keys.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::KMS::Key
- **AWS CloudFormation guard rule:** [CT.KMS.PR.1 rule specification](p. 894)

Details and examples
For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.KMS.PR.1 rule specification (p. 894)

For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.KMS.PR.1 example templates (p. 896)

Explanation

Key rotation minimizes the possibility of key exposure to malicious users. Cryptographic best practices discourage extensive reuse of encryption keys. Rotation of keys on regular basis helps you meet organizational security and compliance requirements.

Usage considerations

- This control applies only to AWS KMS symmetric-encryption, customer managed keys.

Remediation for rule failure

Set EnableKeyRotation to true for AWS KMS symmetric-encryption keys.

The examples that follow show how to implement this remediation.

AWS KMS key - Example

AWS KMS customer managed key configured with key rotation activated. The example is shown in JSON and in YAML.

JSON example

```

{
    "KMSKey": {
        "Type": "AWS::KMS::Key",
        "Properties": {
            "PendingWindowInDays": 7,
            "KeyPolicy": {
                "Version": "2012-10-17",
                "Id": "sample-policy",
                "Statement": [
                    {
                        "Sid": "Enable IAM User Permissions",
                        "Effect": "Allow",
                        "Principal": {
                            "AWS": {
                                "Fn::Sub": "arn:${AWS::Partition}:iam::${AWS::AccountId}:root"
                            }
                        },
                        "Action": "kms:*",
                        "Resource": "*"
                    }
                ]
            }
        }
    }
}
```

YAML example

```

```
KMSKey:
  Type: AWS::KMS::Key
  Properties:
    - PendingWindowInDays: 7
    - KeyPolicy:
      - Version: 2012-10-17
      - Id: sample-policy
      - Statement:
        - Sid: Enable IAM User Permissions
          - Effect: Allow
          - Principal:
            - AWS: !Sub 'arn:${AWS::Partition}:iam:${AWS::AccountId}:root'
          - Action: kms:*
          - Resource: '*'
          - EnableKeyRotation: true

CT.KMS.PR.1 rule specification

# ###################################################################
##       Rule Specification        ##
# Rule Identifier:
#   kms_key_rotation_enabled_check
# Description:
#   This control checks whether key rotation is enabled for AWS KMS customer managed keys.
# Reports on:
#   AWS::KMS::Key
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
#   None
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any KMS key resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a KMS key resource
#     And: 'KeySpec' is provided and is a value other than 'SYMMETRIC_DEFAULT'
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a KMS key resource
#     And: 'KeySpec' is not provided or is provided and is set to 'SYMMETRIC_DEFAULT'
#     And: 'EnableKeyRotation' is not provided
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a KMS key resource

And: 'KeySpec' is not provided or is provided and is set to 'SYMMETRIC_DEFAULT'
And: 'EnableKeyRotation' is provided and is set to bool(false)
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a KMS key resource
And: 'KeySpec' is not provided or is provided and is set to 'SYMMETRIC_DEFAULT'
And: 'EnableKeyRotation' is provided and is set to bool(true)
Then: PASS

Constants
let KMS_KEY_TYPE = "AWS::KMS::Key"
let INPUT_DOCUMENT = this

Assignments
let kms_keys = Resources.*[ Type == %KMS_KEY_TYPE ]

Primary Rules
rule kms_key_rotation_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%kms_keys not empty {
  check(%kms_keys.Properties)
  %kms_keys not empty {
    check(<<
      [CT.KMS.PR.1]: Require any AWS KMS key to have rotation configured
      [FIX]: Set 'EnableKeyRotation' to 'true' for AWS KMS symmetric-encryption keys.
    >>)
  }
}
rule kms_key_rotation_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %KMS_KEY_TYPE) {
  check(%INPUT_DOCUMENT.%KMS_KEY_TYPE.resourceProperties)
  check(<<
    [CT.KMS.PR.1]: Require any AWS KMS key to have rotation configured
    [FIX]: Set 'EnableKeyRotation' to 'true' for AWS KMS symmetric-encryption keys.
  >>)
}

Parameterized Rules
rule check(kms_keys) {
  %kms_keys{
    # Scenario 2
    filter_is_kms_cmk_symmetric_key(this)
  }
  { # Scenario 3, 4 and 5
    EnableKeyRotation exists
    EnableKeyRotation == true
  }
}
rule filter_is_kms_cmk_symmetric_key(kms_key) {
  %kms_key{
    KeySpec not exists or
    KeySpec == "SYMMETRIC_DEFAULT"
  }
}

Utility Rules
CT.KMS.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
Key:
  Type: AWS::KMS::Key
Properties:
  PendingWindowInDays: 7
  KeyPolicy:
    Version: 2012-10-17
    Id: example-policy
    Statement:
      - Sid: Enable IAM User Permissions
        Effect: Allow
        Principal:
          AWS:
            Fn::Sub: arn:${AWS::Partition}:iam::${AWS::AccountId}:root
        Action: kms:*
        Resource: '*'
    EnableKeyRotation: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
Key:
  Type: AWS::KMS::Key
Properties:
  PendingWindowInDays: 7
  KeyPolicy:
    Version: 2012-10-17
    Id: example-policy
    Statement:
      - Sid: Enable IAM User Permissions
        Effect: Allow
        Principal:
          AWS:
            Fn::Sub: arn:${AWS::Partition}:iam::${AWS::AccountId}:root
        Action: kms:*
        Resource: '*'
    EnableKeyRotation: false
[CT.KMS.PR.2] Require that an AWS Key Management Service asymmetric key with RSA key material used for encryption has a key length greater than 2048 bits

This control checks whether an AWS KMS asymmetric key with RSA key material, which is used for encryption and decryption, to use a key spec with a key length greater than 2048 bits (that is, a key spec other than RSA_2048).

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::KMS::Key
- **AWS CloudFormation guard rule:** CT.KMS.PR.2 rule specification (p. 898)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.KMS.PR.2 rule specification (p. 898)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.KMS.PR.2 example templates (p. 900)

**Explanation**

AWS Control Tower recommends using an RSA key spec with a key length greater than 2048 bits, when you are using such keys for encryption and decryption. The key spec determines whether the KMS key is symmetric or asymmetric. It also determines the type of key material, and the algorithms it supports. AWS KMS supports asymmetric KMS keys that represent a mathematically-related RSA or elliptic curve (ECC) public and private key pair. A KMS key with an RSA key pair can be used for encryption and decryption, or for signing and verification (but not both). AWS KMS supports several key lengths for different security requirements.

**Usage considerations**

- This control applies only to a KMS key with an RSA key spec, which is configured for encryption and decryption.

**Remediation for rule failure**

For KMS keys with an RSA keyspec, which are configured for encryption and decryption (KeyUsage of ENCRYPT_DECRYPT), set the KeySpec parameter to a key spec other than RSA_2048.

The examples that follow show how to implement this remediation.

**AWS KMS key - Example**

An AWS KMS asymmetric key configured for encryption and decryption, with an RSA_4096 key spec. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "Key": {
    "Type": "AWS::KMS::Key",
    "KeySpec": "RSA_4096"
  }
}
```
"Properties": {
    "KeyPolicy": {
        "Version": "2012-10-17",
        "Id": "example-policy",
        "Statement": [
            {
                "Sid": "Enable IAM User Permissions",
                "Effect": "Allow",
                "Principal": {
                    "AWS": {
                        "Fn::Sub": "arn:${AWS::Partition}:iam::${AWS::AccountId}:root"
                    }
                },
                "Action": "kms:*",
                "Resource": "*"
            }
        ],
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeySpec": "RSA_4096"
    }
}

YAML example

Key: 
  Type: AWS::KMS::Key
Properties: 
  KeyPolicy: 
    Version: '2012-10-17'
    Id: example-policy
    Statement: 
      - Sid: Enable IAM User Permissions
        Effect: Allow
        Principal: 
          AWS: !Sub 'arn:${AWS::Partition}:iam::${AWS::AccountId}:root'
        Action: kms:*
        Resource: '*
        KeyUsage: ENCRYPT_DECRYPT
        KeySpec: RSA_4096

CT.KMS.PR.2 rule specification

# ####################################################################
# Rule Specification    ##
# ####################################################################

# Rule Name:
# kms_asymmetric_rsa_keyspec_check
#
# Description:
# This control checks whether an AWS KMS asymmetric key with RSA key material, which is
# used for encryption and decryption, to use a key spec with a key length greater than 2048
# bits (that is, a key spec other than 'RSA_2048').
#
# Reports on:
# AWS::KMS::Key
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
# None
# Scenarios:
  # Scenario: 1
  # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
  # And: The input document does not contain any KMS key resources
  # Then: SKIP
  # Scenario: 2
  # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
  # And: The input document contains a KMS key resource
  # And: 'KeyUsage' has been provided and is a value other than 'ENCRYPT_DECRYPT'
  # Then: SKIP
  # Scenario: 3
  # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
  # And: The input document contains a KMS key resource
  # And: 'KeyUsage' has not been provided or has been provided and is set to
  # 'ENCRYPT_DECRYPT'
  # And: 'KeySpec' has not been provided or has been provided and is set to a
  # key spec other than an RSA key spec (does not begin with 'RSA_')
  # Then: SKIP
  # Scenario: 4
  # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
  # And: The input document contains a KMS key resource
  # And: 'KeyUsage' has not been provided or has been provided and is set to
  # 'ENCRYPT_DECRYPT'
  # And: 'KeySpec' has been provided and is set to an RSA key spec (begins with 'RSA_')
  # And: 'KeySpec' has been set to a disallowed RSA key spec ('RSA_2048')
  # Then: FAIL
  # Scenario: 5
  # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
  # And: The input document contains a KMS key resource
  # And: 'KeyUsage' has not been provided or has been provided and is set to
  # 'ENCRYPT_DECRYPT'
  # And: 'KeySpec' has been provided and is set to an RSA key spec (begins with 'RSA_')
  # And: 'KeySpec' has not been set to a disallowed RSA key spec ('RSA_2048')
  # Then: PASS

# Constants

let KMS_KEY_TYPE = "AWS::KMS::Key"
let RSA_KEYSPEC_PATTERN = /^RSA_/$
let ENCRYPTION_KEY_USAGE = "ENCRYPT_DECRYPT"
let DISALLOWED_RSA_KEYSPECs = [ "RSA_2048" ]
let INPUT_DOCUMENT = this

# Assignments

let kms_keys = Resources.*[ Type == %KMS_KEY_TYPE ]

# Primary Rules

rule kms_asymmetric_rsa_keyspec_check when is_cfn_template(%INPUT_DOCUMENT)
%kms_keys not empty {
    <<
    [CT.KMS.PR.2]: Require that an AWS KMS asymmetric key with RSA key material used for encryption has a key length greater than 2048 bits
    [FIX]: For KMS keys with an RSA keyspec, which are configured for encryption and decryption ('KeyUsage' of 'ENCRYPT_DECRYPT'), set the 'KeySpec' parameter to a key spec other than 'RSA_2048'.
    >>
}

rule kms_asymmetric_rsa_keyspec_check when is_cfn_hook(%INPUT_DOCUMENT, %KMS_KEY_TYPE) {
    check(%INPUT_DOCUMENT.%KMS_KEY_TYPE.resourceProperties)
    <<
    [CT.KMS.PR.2]: Require that an AWS KMS asymmetric key with RSA key material used for encryption has a key length greater than 2048 bits
    [FIX]: For KMS keys with an RSA keyspec, which are configured for encryption and decryption ('KeyUsage' of 'ENCRYPT_DECRYPT'), set the 'KeySpec' parameter to a key spec other than 'RSA_2048'.
    >>
}

# Parameterized Rules
#
rule check(kms_keys) {
    %kms_keys[
        # Scenarios 2 and 3
        filter_is_kms_rsa_asymmetric_encryption_key(this)
    ] {
        # Scenario 4 and 5
        KeySpec exists
        KeySpec not in %DISALLOWED_RSA_KEYSPECS
    }
}

rule filter_is_kms_rsa_asymmetric_encryption_key(kms_key) {
    %kms_key {
        KeyUsage not exists or
        KeyUsage == %ENCRYPTION_KEY_USAGE
        KeySpec exists
        KeySpec == %RSA_KEYSPEC_PATTERN
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.KMS.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.
PASS Example - Use this template to verify a compliant resource creation.

Resources:
Key:
  Type: AWS::KMS::Key
Properties:
  KeyPolicy:
    Version: '2012-10-17'
    Id: example-policy
    Statement:
      - Sid: Enable IAM User Permissions
        Effect: Allow
        Principal:
          AWS:
            Fn::Sub: arn:${AWS::Partition}:iam::${AWS::AccountId}:root
        Action: kms:*
        Resource: '*'
        KeyUsage: ENCRYPT_DECRYPT
        KeySpec: RSA_4096

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
Key:
  Type: AWS::KMS::Key
Properties:
  KeyPolicy:
    Version: '2012-10-17'
    Id: example-policy
    Statement:
      - Sid: Enable IAM User Permissions
        Effect: Allow
        Principal:
          AWS:
            Fn::Sub: arn:${AWS::Partition}:iam::${AWS::AccountId}:root
        Action: kms:*
        Resource: '*'
        KeyUsage: ENCRYPT_DECRYPT
        KeySpec: RSA_2048

[CT.KMS.PR.3] Require an AWS Key Management Service key policy to have a statement that limits creation of AWS KMS grants to AWS services

This control checks whether an AWS KMS key has an associated key policy statement that limits creation of AWS KMS grants to AWS services only.

- **Control objective:** Enforce least privilege
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::KMS::Key
- **AWS CloudFormation guard rule:** CT.KMS.PR.3 rule specification (p. 903)

Details and examples
Explanation

Users with permission to create grants for a KMS key (kms:CreateGrant) can use a grant to allow other users and roles, including AWS services, to use the KMS key (grantee principals). Grantee principals can be identities in your own AWS account, or identities from a different account or organization.

By denying creation of AWS KMS grants unless the request originates from an AWS service, you prevent grants from being assigned directly to principals other than AWS service principals, and you reduce the opportunities for grant misuse. The kms:GrantIsForAWSResource condition helps check whether the CreateGrant operation is being called by an AWS service integrated with AWS KMS, on behalf of another principal. The aws:PrincipalIsAWSService condition helps check whether the CreateGrant operation is being called directly by an AWS service principal.

Usage considerations

- If you must use additional conditions on your grants, or if you must issue AWS KMS grants directly to your IAM principals for a customer-managed key, do not enable this control. This control requires a policy statement that denies the creation of AWS KMS grants for your customer-managed KMS keys, if the request does not originate from an AWS service that's integrated with AWS KMS, or from an AWS service principal.

Remediation for rule failure

Configure an AWS KMS policy statement that denies access to the kms:CreateGrant operation for all principals when the kms:GrantIsForAWSResource and aws:PrincipalIsAWSService conditions are both false, using the BoolIfExists condition operator.

The examples that follow show how to implement this remediation.

AWS KMS key - Example

An AWS KMS key, configured to deny creation of AWS KMS grants where the CreateGrant request does not originate from an AWS service principal. The example is shown in JSON and in YAML.

JSON example

```json
{
    "Key": {
        "Type": "AWS::KMS::Key",
        "Properties": {
            "KeyUsage": "ENCRYPT_DECRYPT",
            "KeySpec": "SYMMETRIC_DEFAULT",
            "KeyPolicy": {
                "Version": "2012-10-17",
                "Id": "sample-policy",
                "Statement": [
                    {
                        "Sid": "Enable IAM User Permissions",
                        "Effect": "Allow",
                        "Principal": {
                            "AWS": {
                                "Fn::Sub": "arn:${AWS::Partition}:iam::${AWS::Account}:root"
                            }
                        }
                    }
                ]
            }
        }
    }
}
```
YAML example

Key:
  Type: AWS::KMS::Key
Properties:
  KeyUsage: ENCRYPT_DECRYPT
  KeySpec: SYMMETRIC_DEFAULT
  KeyPolicy:
    Version: '2012-10-17'
    Id: sample-policy
    Statement:
      - Sid: Enable IAM User Permissions
        Effect: Allow
        Principal:
          AWS: !Sub 'arn:${AWS::Partition}:iam:${AWS::AccountId}:root'
        Action: kms:*
        Resource: '*'
      - Effect: Deny
        Action: kms:CreateGrant
        Resource: '*'
        Principal: '*'
        Condition:
          BoolIfExists:
            kms:GrantIsForAWSResource: 'false'
            aws:PrincipalIsAWSService: 'false'

CT.KMS.PR.3 rule specification

```yaml
# ###################################
##       Rule Specification        ##
#####################################
#
# Rule Name:
#  kms_create_grant_aws_service_check
#
```
# Description:
# This control checks whether an AWS KMS key has an associated key policy statement that
# limits creation of AWS KMS grants to AWS services only.
#
# Reports on:
# AWS::KMS::Key
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document does not contain any KMS key resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a KMS key resource
# And: 'Statement' in 'KeyPolicy' has not been provided or has been provided as an
# empty list
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a KMS key resource
# And: 'Statement' in 'KeyPolicy' has been provided as a non-empty list
# And: 'Statement' in 'KeyPolicy' does not include a statement that denies all
# Principals ('*', AWS: '*')
# create grant permissions ('kms:CreateGrant') on the KMS key (resource of '*')
# when the conditions 'kms:GrantIsForAWSResource' and
# 'aws:PrincipalIsAWSService'
# are both 'false' ('BoolIfExists' condition operator)
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a KMS key resource
# And: 'Statement' in 'KeyPolicy' has been provided as a non-empty list
# And: 'Statement' in 'KeyPolicy' includes a statement that denies all Principals
# ('*', AWS: '*')
# create grant permissions ('kms:CreateGrant') on the KMS key (resource of '*')
# when the conditions 'kms:GrantIsForAWSResource' and
# 'aws:PrincipalIsAWSService'
# are both 'false' ('BoolIfExists' condition operator)
# Then: PASS
#
# Constants
#
let INPUT_DOCUMENT = this
let KMS_KEY_TYPE = "AWS::KMS::Key"
let KMS_GRANT_IS_FOR_AWS_RESOURCE_KEY_PATTERN = /^(?i)kms:GrantIsForAWSResource$/
let AWS_PRINCIPAL_IS_AWS_SERVICE_KEY_PATTERN = /^(?i)aws:PrincipalIsAWSService$/
let ALLOWED_KEY_PATTERNS = [ /^(?i)kms:GrantIsForAWSResource$/, /^(?i)aws:PrincipalIsAWSService$/]
#
# Assignments
#
let kms_keys = Resources.*[ Type == %KMS_KEY_TYPE ]
# Primary Rules

rule kms_create_grant_aws_service_check when is_cfn_template(%INPUT_DOCUMENT) %kms_keys not empty {
    check(%kms_keys.Properties)
    
    <<
    [CT.KMS.PR.3]: Require an AWS KMS key policy to have a statement that limits creation of AWS KMS grants to AWS services
    [FIX]: Configure a KMS key policy statement that denies access to the 'kms:CreateGrant' operation for all principals when the 'kms:GrantIsForAWSResource' and 'aws:PrincipalIsAWSService' conditions are both false, using the 'BoolIfExists' condition operator.
    >>
}

rule kms_create_grant_aws_service_check when is_cfn_hook(%INPUT_DOCUMENT, %KMS_KEY_TYPE) {
    check(%INPUT_DOCUMENT.%KMS_KEY_TYPE.resourceProperties)
    
    <<
    [CT.KMS.PR.3]: Require an AWS KMS key policy to have a statement that limits creation of AWS KMS grants to AWS services
    [FIX]: Configure a KMS key policy statement that denies access to the 'kms:CreateGrant' operation for all principals when the 'kms:GrantIsForAWSResource' and 'aws:PrincipalIsAWSService' conditions are both false, using the 'BoolIfExists' condition operator.
    >>
}

# Parameterized Rules

rule check(kms_keys) {
    %kms_keys {
        # Scenario 2
        KeyPolicy exists
        KeyPolicy is_struct

        KeyPolicy {
            Statement exists
            Statement is_list
            Statement not empty

            #Scenario 3 and 4
            some Statement[*] {
                check_statement_create_grant_aws_services_only(this)
            }
        }
    }
}

rule check_statement_create_grant_aws_services_only(statement) {
    %statement{
        check_all_required_statement_properties(this)

        Effect == "Deny"
        Action[*] in ["kms:CreateGrant"]

        Principal == "*" or
        Principal {
            AWS exists
            AWS == "*"
        }

        Resource[*] == "*"

        Condition is_struct
        struct_contains_only_allowed_keys(Condition, ["BoolIfExists"])
    }
}
CT.KMS.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  Key:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Amazon Kinesis controls

Topics

- [CT.KINESIS.PR.1] Require any Amazon Kinesis data stream to have encryption at rest configured (p. 907)

[CT.KINESIS.PR.1] Require any Amazon Kinesis data stream to have encryption at rest configured

This control checks whether Amazon Kinesis data streams are encrypted at rest with server-side encryption.
- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Kinesis::Stream
- **AWS CloudFormation guard rule:** [CT.KINESIS.PR.1 rule specification (p. 909)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.KINESIS.PR.1 rule specification (p. 909)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.KINESIS.PR.1 example templates (p. 911)]

**Explanation**

Server-side encryption is a feature in Amazon Kinesis data streams that encrypts data automatically, before the data is at rest, by using an AWS KMS key. Data is encrypted before it is written to the Kinesis stream storage layer, and decrypted after it is retrieved from storage. As a result, your data is encrypted at rest within the Amazon Kinesis data stream service.

**Remediation for rule failure**

Specify a `StreamEncryption` configuration, with `EncryptionType` set to `KMS` and `KeyId` set to an AWS KMS key identifier.

The examples that follow show how to implement this remediation.

**Amazon Kinesis Data Stream - Example**

Amazon Kinesis data stream configured to encrypt data at rest with server-side encryption, using an AWS KMS key. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "KinesisStream": {
        "Type": "AWS::Kinesis::Stream",
        "Properties": {
            "RetentionPeriodHours": 168,
            "ShardCount": 3,
            "StreamEncryption": {
                "EncryptionType": "KMS",
                "KeyId": {
                    "Ref": "KMSKey"
                }
            }
        }
    }
}
```

**YAML example**

```yaml
KinesisStream:
  Type: AWS::Kinesis::Stream
```
Properties:
  - RetentionPeriodHours: 168
  - ShardCount: 3
  - StreamEncryption:
    - EncryptionType: KMS
    - KeyId: !Ref 'KMSKey'

CT.KINESIS.PR.1 rule specification

```plaintext
# #####################################################################
##       Rule Specification        ##
#####################################################################
# Rule Identifier:
#   kinesis_stream_encrypted_check
#
# Description:
#   This control checks whether Amazon Kinesis data streams are encrypted at rest with
#   server-side encryption.
#
# Reports on:
#   AWS::Kinesis::Stream
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any Kinesis stream resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an Kinesis stream resource
#     And: 'StreamEncryption' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an Kinesis stream resource
#     And: 'StreamEncryption' has been provided
#     And: 'StreamEncryption.EncryptionType' has not been provided or provided as an
#     empty string
#     And: 'StreamEncryption.KeyId' has not been provided or provided as an empty string
#     or invalid local reference
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an Kinesis stream resource
#     And: 'StreamEncryption' has been provided
#     And: 'StreamEncryption.EncryptionType' has been provided as a non-empty string
#     And: 'StreamEncryption.KeyId' has not been provided or provided as an empty string
#     or invalid local reference
#     Then: FAIL
#   Scenario: 5
```
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#    And: The input document contains an Kinesis stream resource
#    And: 'StreamEncryption' has been provided
#    And: 'StreamEncryption.EncryptionType' has not been provided or provided as an
#      empty string
#    And: 'StreamEncryption.KeyId' has been provided as a non-empty string or valid
#      local reference
#    Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#    And: The input document contains an Kinesis stream resource
#    And: 'StreamEncryption' has been provided
#    And: 'StreamEncryption.EncryptionType' has been provided as a non-empty string
#    And: 'StreamEncryption.KeyId' has been provided as a non-empty string or valid
#      local reference
#    Then: PASS
#
# Constants
#
let KINESIS_STREAM_TYPE = "AWS::Kinesis::Stream"
let INPUT_DOCUMENT = this
#
# Assignments
#
let kinesis_streams = Resources.*[ Type == %KINESIS_STREAM_TYPE ]
#
# Primary Rules
#
rule kinesis_stream_encrypted_check when is_cfn_template(%INPUT_DOCUMENT)
    %kinesis_streams not empty {
        check(%kinesis_streams.Properties)
        <<
        [CT.KINESIS.PR.1]: Require any Amazon Kinesis data stream to have encryption at
        rest configured
[FIX]: Specify a 'StreamEncryption' configuration, with 'EncryptionType' set to
'KMS' and 'KeyId' set to an AWS KMS key identifier.
        >>
    }

rule kinesis_stream_encrypted_check when is_cfn_hook(%INPUT_DOCUMENT, %KINESIS_STREAM_TYPE) {
    check(%INPUT_DOCUMENT.%KINESIS_STREAM_TYPE.resourceProperties)
    <<
    [CT.KINESIS.PR.1]: Require any Amazon Kinesis data stream to have encryption at
    rest configured
    [FIX]: Specify a 'StreamEncryption' configuration, with 'EncryptionType' set to
'KMS' and 'KeyId' set to an AWS KMS key identifier.
    >>
}
#
# Parameterized Rules
#
rule check(kinesis_stream) {
    %kinesis_stream {
        # Scenario 2
        StreamEncryption exists
        StreamEncryption is_struct

        StreamEncryption {
            # Scenario 3
CT.KINESIS.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
KMSKey:
  Type: AWS::KMS::Key
Properties:
  PendingWindowInDays: 7
KeyPolicy:
  Version: 2012-10-17
  Id: example-key-policy
  Statement:
    - Sid: Enable IAM User Permissions
      Effect: Allow
      Principal:
        AWS:
          Fn::Sub: arn:${AWS::Partition}:iam::${AWS::AccountId}:root
        Action: kms:*
        Resource: '*'
      KeySpec: SYMMETRIC_DEFAULT
KinesisStream:
  Type: AWS::Kinesis::Stream
  Properties:
    RetentionPeriodHours: 168
    ShardCount: 3
  StreamEncryption:
    EncryptionType: KMS
    KeyId:
      Ref: KMSKey

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
KinesisStream:
  Type: AWS::Kinesis::Stream
  Properties:
    RetentionPeriodHours: 168
    ShardCount: 3

AWS Lambda controls

Topics
- [CT.LAMBDA.PR.2] Require AWS Lambda function policies to prohibit public access (p. 912)
- [CT.LAMBDA.PR.3] Require an AWS Lambda function to be in a customer-managed Amazon Virtual Private Cloud (VPC) (p. 920)
- [CT.LAMBDA.PR.4] Require an AWS Lambda layer permission to grant access to an AWS organization or specific AWS account (p. 926)
- [CT.LAMBDA.PR.5] Require an AWS Lambda function URL to use AWS IAM-based authentication (p. 931)
- [CT.LAMBDA.PR.6] Require an AWS Lambda function URL CORS policy to restrict access to specific origins (p. 936)

[CT.LAMBDA.PR.2] Require AWS Lambda function policies to prohibit public access

This control checks whether an AWS Lambda function resource-based policy prohibits public access.

- Control objective: Limit network access
- Implementation: AWS CloudFormation Guard Rule
• **Control behavior:** Proactive
• **Resource types:** AWS::Lambda::Permission
• **AWS CloudFormation guard rule:** [CT.LAMBDA.PR.2 rule specification (p. 915)](#)

**Details and examples**

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.LAMBDA.PR.2 rule specification (p. 915)](#)

• For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.LAMBDA.PR.2 example templates (p. 918)](#)

**Explanation**

The Lambda function should not be publicly accessible, because it may permit unintended access to your code stored in the function.

**Remediation for rule failure**

When setting `Principal` to *, provide one of `SourceAccount`, `SourceArn`, or `PrincipalOrgID`. When setting `Principal` to a service principal (for example, s3.amazonaws.com), provide one of `SourceAccount` or `SourceArn`.

The examples that follow show how to implement this remediation.

**AWS Lambda Function Policy - Example One**

AWS Lambda function policy configured with an AWS account ID principal. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "LambdaPermission": {
        "Type": "AWS::Lambda::Permission",
        "Properties": {
            "Action": "lambda:InvokeFunction",
            "FunctionName": {
                "Ref": "LambdaFunction"
            },
            "Principal": {
                "Ref": "AWS::AccountId"
            }
        }
    }
}
```

**YAML example**

```
LambdaPermission:
  Type: AWS::Lambda::Permission
  Properties:
    Action: lambda:InvokeFunction
    FunctionName: !Ref 'LambdaFunction'
    Principal: !Ref 'AWS::AccountId'
```
The examples that follow show how to implement this remediation.

AWS Lambda Function Policy - Example Two

AWS Lambda function policy configured with a wildcard principal and source account condition. The example is shown in JSON and in YAML.

**JSON example**

```
{  
  "LambdaPermission": {  
    "Type": "AWS::Lambda::Permission",  
    "Properties": {  
      "Action": "lambda:InvokeFunction",  
      "FunctionName": {  
        "Ref": "LambdaFunction"
      },  
      "Principal": "*",  
      "SourceAccount": {  
        "Ref": "AWS::AccountId"
      }
    }
  }
}
```

**YAML example**

```
LambdaPermission:
  Type: AWS::Lambda::Permission
  Properties:
    Action: lambda:InvokeFunction
    FunctionName: !Ref 'LambdaFunction'
    Principal: '*'
    SourceAccount: !Ref 'AWS::AccountId'
```

The examples that follow show how to implement this remediation.

AWS Lambda Function Policy - Example Three

AWS Lambda function policy configured with a service principal and source ARN condition. The example is shown in JSON and in YAML.

**JSON example**

```
{  
  "LambdaPermission": {  
    "Type": "AWS::Lambda::Permission",  
    "Properties": {  
      "Action": "lambda:InvokeFunction",  
      "FunctionName": {  
        "Ref": "LambdaFunction"
      },  
      "Principal": "s3.amazonaws.com",  
      "SourceArn": {  
        "Fn::GetAtt": [  
          "S3Bucket",  
          "Arn"
        ]
      }
    }
  }
}
```
YAML example

LambdaPermission:
  Type: AWS::Lambda::Permission
  Properties:
    Action: lambda:InvokeFunction
    FunctionName: !Ref 'LambdaFunction'
    Principal: s3.amazonaws.com
    SourceArn: !GetAtt 'S3Bucket.Arn'

CT.LAMBDA.PR.2 rule specification

# ###############################
##       Rule Specification    ##
# ###############################
#
# Rule Identifier:
# lambda_function_public_access_prohibited_check
#
# Description:
# This control checks whether an AWS Lambda function resource-based policy prohibits public access.
#
# Reports on:
# AWS::Lambda::Permission
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document does not contain any Lambda permission resources
#   Then: SKIP
# Scenario: 2
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains a Lambda permission resource
#   And: 'FunctionUrlAuthType' has been provided with a value of 'NONE'
#   Then: FAIL
# Scenario: 3
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains a Lambda permission resource
#   And: 'Principal' has been provided with a wildcard value ('*')
#   And: 'SourceAccount' has not been provided or provided with an empty string value
#   And: 'SourceArn' has not been provided or provided with an empty string value or non-valid local reference
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a Lambda permission resource
# And: 'Principal' has been provided with value that does not match an AWS Account ID, AWS IAM ARN or wildcard value (**)
# And: 'SourceAccount' has not been provided or provided with an empty string value
# And: 'SourceArn' has not been provided or provided with an empty string value or non-valid local reference
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a Lambda permission resource
# And: 'Principal' has been provided with an AWS Account ID or AWS IAM ARN value
# Then: PASS
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a Lambda permission resource
# And: 'Principal' has been provided with a wildcard value (**)
# And: At least one of 'SourceAccount', 'SourceArn' or 'PrincipalOrgID' have been provided with non-empty string values (or a valid local reference for 'SourceArn')
# Then: PASS
# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a Lambda permission resource
# And: 'Principal' has been provided with value that does not match an AWS Account ID or AWS IAM ARN
# And: At least one of 'SourceAccount', 'SourceArn' have been provided with non-empty string values (or a valid local reference for 'SourceArn')
# Then: PASS

# Constants
let LAMBDA_PERMISSION_TYPE = "AWS::Lambda::Permission"
let AWS_ACCOUNT_ID_PATTERN = /\d{12}/
let AWS_IAM_PRINCIPAL_PATTERN = /arn:aws[a-z0-9-]*:iam::\d{12}:.*/

let INPUT_DOCUMENT = this

# Assignments
let lambda_permissions = Resources.*[ Type == %LAMBDA_PERMISSION_TYPE ]

# Primary Rules
rule lambda_function_public_access_prohibited_check when is_cfn_template(%INPUT_DOCUMENT)
%lambda_permissions not empty {
  check(%lambda_permissions.Properties)
  [CT.LAMBDA.PR.2]: Require AWS Lambda function policies to prohibit public access
  [FIX]: When setting 'Principal' to '**', provide one of 'SourceAccount', 'SourceArn', or 'PrincipalOrgID'. When setting 'Principal' to a service principal (for example, s3.amazonaws.com), provide one of 'SourceAccount' or 'SourceArn'.
}

916
rule lambda_function_public_access_prohibited_check when is_cfn_hook(%INPUT_DOCUMENT, %LAMBDA_PERMISSION_TYPE) {
  check(%INPUT_DOCUMENT.%LAMBDA_PERMISSION_TYPE.resourceProperties)
  <<
    [CT.LAMBDA.PR.2]: Require AWS Lambda function policies to prohibit public access
    [FIX]: When setting 'Principal' to '*', provide one of 'SourceAccount', 'SourceArn', or 'PrincipalOrgID'. When setting 'Principal' to a service principal (for example, s3.amazonaws.com), provide one of 'SourceAccount' or 'SourceArn'.
  >>
}  

# Parameterized Rules

# rule check(lambda_permission) {
%lambda_permission {
  # Scenario 2 and 5
  FunctionUrlAuthType not exists or
  FunctionUrlAuthType != "NONE"
}

%lambda_permission [
  Principal exists
  Principal == "**"
] {
  # Scenario 3 and 6
  SourceAccount exists or
  SourceArn exists or
  PrincipalOrgID exists

  check_is_string_and_not_empty(SourceAccount) or
  check_is_string_or_local_reference(SourceArn) or
  check_is_string_and_not_empty(PrincipalOrgID)
}

%lambda_permission [
  Principal exists
  Principal != "**"
  Principal != %AWS_ACCOUNT_ID_PATTERN
  Principal != %AWS_IAM_PRINCIPAL_PATTERN
] {
  # Scenario 4 and 7
  SourceAccount exists or
  SourceArn exists

  check_is_string_and_not_empty(SourceAccount) or
  check_is_string_or_local_reference(SourceArn)
}
}

rule check_is_string_or_local_reference(value) {
  %value {
    check_is_string_and_not_empty(this) or
    check_local_references(%INPUT_DOCUMENT, this)
  }
}

rule check_local_references(doc, reference_properties) {
  %reference_properties {
    'Fn::GetAtt' {
      query_for_resource(%doc, this[0])
      <<Local Stack reference was invalid>>
    } or Ref {
      query_for_resource(%doc, this)
      <<Local Stack reference was invalid>>
  }
}
CT.LAMBDA.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
LambdaFunctionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service:
            - lambda.amazonaws.com
          Action:
            - sts:AssumeRole
          Path: /
        - PolicyName: LambdaFunctionPolicy
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Effect: Allow
                Action:
                  - logs:CreateLogGroup
                  - logs:CreateLogStream
                  - logs:PutLogEvents
                Resource: '*'
LambdaFunction:
   Type: AWS::Lambda::Function
   Properties:
       Role:
           Fn::GetAtt: LambdaFunctionRole.Arn
       Handler: index.handler
       Runtime: python3.9
       Code:
           ZipFile: "def handler(event, context):
                 print("hello")\n"
       Description: TestS3EventFunction
LambdaPermission:
   Type: AWS::Lambda::Permission
   Properties:
       Action: lambda:InvokeFunction
       FunctionName:
           Ref: LambdaFunction
       Principal:
           Ref: AWS::AccountId

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
LambdaFunctionRole:
   Type: AWS::IAM::Role
   Properties:
       AssumeRolePolicyDocument:
           Version: '2012-10-17'
           Statement:
               - Effect: Allow
                 Principal:
                     Service:
                         - lambda.amazonaws.com
                 Action:
                     - sts:AssumeRole
           Path: /
       Policies:
           - PolicyName: LambdaFunctionPolicy
             PolicyDocument:
                 Version: '2012-10-17'
                 Statement:
                     - Effect: Allow
                       Action:
                           - logs:CreateLogGroup
                           - logs:CreateLogStream
                           - logs:PutLogEvents
                       Resource: '*'
LambdaFunction:
   Type: AWS::Lambda::Function
   Properties:
       Role:
           Fn::GetAtt: LambdaFunctionRole.Arn
       Handler: index.handler
       Runtime: python3.9
       Code:
           ZipFile: "def handler(event, context):
                 print("hello")\n"
       Description: TestS3EventFunction
LambdaPermission:
   Type: AWS::Lambda::Permission
   Properties:
       Action: lambda:InvokeFunction
       FunctionName:
           Ref: LambdaFunction
Proactive controls

[CT.LAMBDA.PR.3] Require an AWS Lambda function to be in a customer-managed Amazon Virtual Private Cloud (VPC)

This control checks whether an AWS Lambda function has been configured with access to resources in a customer-managed Amazon Virtual Private Cloud (VPC).

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Lambda::Function

**AWS CloudFormation guard rule:** [CT.LAMBDA.PR.3 rule specification](p. 921)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.LAMBDA.PR.3 rule specification](p. 921)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.LAMBDA.PR.3 example templates](p. 924)

**Explanation**

AWS Lambda functions can be linked to private subnets within a virtual private cloud (VPC) in your AWS account to connect to resources such as databases, cache instances, or internal services. Ensure that the subnets and security groups used allow access to the necessary resources.

**Usage considerations**

- This control does not evaluate the VPC subnet routing configuration to determine public reachability.
- This control does not support AWS Lambda@Edge Functions. Lambda@Edge does not support functions that are configured with access to resources inside your VPC.
- Lambda functions can't connect directly to a VPC with dedicated instance tenancy. To connect to resources in a dedicated VPC, peer it to a second VPC with default tenancy.

**Remediation for rule failure**

In VpcConfig, provide the SubnetIds property with one or more Subnet IDs, and provide the SecurityGroupIds property with one or more Security Group IDs.

The examples that follow show how to implement this remediation.

**AWS Lambda Function - Example**

AWS Lambda function configured to access resources in a VPC. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "LambdaFunction": {
        "Type": "AWS::Lambda::Function",
```
"Properties": {
    "Role": {
        "Fn::GetAtt": "LambdaFunctionRole.Arn"
    },
    "Handler": "index.handler",
    "Code": {
        "ZipFile": "def handler(event, context):
            print("sample function")\n"
    },
    "Runtime": "python3.9",
    "VpcConfig": {
        "SubnetIds": [
            {
                "Fn::GetAtt": [
                    "SubnetOne",
                    "SubnetId"
                ]
            },
            {
                "Fn::GetAtt": [
                    "SubnetTwo",
                    "SubnetId"
                ]
            }
        ],
        "SecurityGroupIds": [
            {
                "Fn::GetAtt": [
                    "SecurityGroup",
                    "GroupId"
                ]
            }
        ]
    }
}

YAML example

LambdaFunction:
  Type: AWS::Lambda::Function
  Properties:
    Role: !GetAtt 'LambdaFunctionRole.Arn'
    Handler: index.handler
    Code:
      ZipFile: "def handler(event, context):
            print("sample function")\n"
    Runtime: python3.9
  VpcConfig:
    SubnetIds:
      - !GetAtt 'SubnetOne.SubnetId'
      - !GetAtt 'SubnetTwo.SubnetId'
    SecurityGroupIds:
      - !GetAtt 'SecurityGroup.GroupId'

CT.LAMBDA.PR.3 rule specification

# ####################################################################
##  Rule Specification  ##
# ####################################################################

# Rule Identifier:
#   lambda_inside_vpc_check

# Description:
#   This control checks whether an AWS Lambda function has been configured with access to
#   resources in a customer-managed Amazon Virtual Private Cloud (VPC).

# Reports on:
#   AWS::Lambda::Function

# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
#   None

# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any Lambda function resources
#     Then: SKIP

# Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a Lambda function resource
#     And: 'VpcConfig' has not been provided
#     Then: FAIL

# Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a Lambda function resource
#     And: 'VpcConfig' has been provided
#     And: 'SubnetIds' in 'VpcConfig' has been provided as a non-empty list that contains
#          non-empty strings or valid
#          local references
#     And: 'SecurityGroupIds' in 'VpcConfig' has not been been provided or has been
#          provided as an empty list
#     Then: FAIL

# Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a Lambda function resource
#     And: 'VpcConfig' has been provided
#     And: 'SecurityGroupIds' in 'VpcConfig' has been provided as a non-empty list that
#          contains non-empty strings
#          or valid local references
#     And: 'SubnetIds' in 'VpcConfig' has not been been provided or has been provided as
#          an empty list
#     Then: FAIL

# Scenario: 5
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a Lambda function resource
#     And: 'VpcConfig' has been provided
#     And: 'SecurityGroupIds' in 'VpcConfig' has been provided as a non-empty list that
#          contains non-empty strings or
#          valid local references
#     And: 'SubnetIds' in 'VpcConfig' has not been been provided or has been provided as
#          non-empty strings or valid
#          local references
#     Then: PASS
# Constants
#
let LAMBDA_FUNCTION_TYPE = "AWS::Lambda::Function"
let INPUT_DOCUMENT = this
#
# Assignments
#
let lambda_functions = Resources.*[ Type == %LAMBDA_FUNCTION_TYPE ]
#
# Primary Rules
#
rule lambda_inside_vpc_check when is_cfn_template(%INPUT_DOCUMENT)
%lambda_functions not empty {
    check(%lambda_functions.Properties)
    <<
    [CT.LAMBDA.PR.3]: Require an AWS Lambda function to be in a customer-managed Amazon Virtual Private Cloud (VPC)
    [FIX]: In 'VpcConfig', provide the 'SubnetIds' property with one or more Subnet IDs, and provide the 'SecurityGroupIds' property with one or more Security Group IDs.
    >>
}
rule lambda_inside_vpc_check when is_cfn_hook(%INPUT_DOCUMENT, %LAMBDA_FUNCTION_TYPE) {
    check(%INPUT_DOCUMENT.%LAMBDA_FUNCTION_TYPE.resourceProperties)
    <<
    [CT.LAMBDA.PR.3]: Require an AWS Lambda function to be in a customer-managed Amazon Virtual Private Cloud (VPC)
    [FIX]: In 'VpcConfig', provide the 'SubnetIds' property with one or more Subnet IDs, and provide the 'SecurityGroupIds' property with one or more Security Group IDs.
    >>
}
#
# Parameterized Rules
#
rule check(lambda_function) {
    %lambda_function {
        # Scenario 2
        VpcConfig exists
        VpcConfig is_struct
        VpcConfig {
            # Scenario 3 and 5
            SubnetIds exists
            SubnetIds is_list
            SubnetIds not empty
            SubnetIds[*] {
                check_is_string_and_not_empty(this) or
                check_local_references(%INPUT_DOCUMENT, this, "AWS::EC2::Subnet")
            }
            # Scenario 4 and 5
            SecurityGroupIds exists
            SecurityGroupIds is_list
            SecurityGroupIds not empty
            SecurityGroupIds[*] {
                check_is_string_and_not_empty(this) or
                check_local_references(%INPUT_DOCUMENT, this, "AWS::EC2::SecurityGroup")
            }
        }
    }
}
#
# Utility Rules

# is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

rule check_is_string_and_not_empty(value) {
    %value {
        this is_string
        this != /\A\s*\z/
    }
}

rule check_local_references(doc, reference_properties, referenced_resource_type) {
    %reference_properties {
        'Fn::GetAtt' {
            query_for_resource(%doc, this[0], %referenced_resource_type)
            <<Local Stack reference was invalid>>
        } or Ref {
            query_for_resource(%doc, this, %referenced_resource_type)
            <<Local Stack reference was invalid>>
        }
    }
}

rule query_for_resource(doc, resource_key, referenced_resource_type) {
    let referenced_resource = %doc.Resources[ keys == %resource_key ]
    %referenced_resource not empty
    %referenced_resource {
        Type == %referenced_resource_type
    }
}

CT.LAMBDA.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
    Type: AWS::EC2::VPC
    Properties:
        CidrBlock: 10.0.0.0/16
        EnableDnsSupport: 'true'
        EnableDnsHostnames: 'true'
SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
        VpcId:
            Ref: VPC
        CidrBlock: 10.0.0.0/24
        AvailabilityZone:
            Fn::Select:
                - 0
SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
      Ref: VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone:
      Fn::Select:
        - 1
      - Fn::GetAZs: ''
SecurityGroup1:
  Type: AWS::EC2::SecurityGroup
  Properties:
    VpcId:
      Ref: VPC
    GroupDescription:
      Fn::Sub: ${AWS::StackName}-example
LambdaFunctionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - lambda.amazonaws.com
          Action:
            - sts:AssumeRole
          Path: /
    Policies:
      - PolicyName: LambdaFunctionPolicy
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - logs:CreateLogGroup
                - logs:CreateLogStream
                - logs:PutLogEvents
                - ec2:CreateNetworkInterface
                - ec2:DescribeNetworkInterfaces
                - ec2:DeleteNetworkInterface
              Resource: '*'
LambdaFunction:
  Type: AWS::Lambda::Function
  Properties:
    Role:
      Fn::GetAtt: LambdaFunctionRole.Arn
    Handler: index.handler
    Code:
      ZipFile: |
        def handler(event, context):
          print("example")
    Runtime: python3.9
VpcConfig:
  SubnetIds:
    - Fn::GetAtt:
      - SubnetOne
      - SubnetId
    - Fn::GetAtt:
      - SubnetTwo
      - SubnetId
  SecurityGroupIds:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
LambdaFunctionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - lambda.amazonaws.com
          Action:
            - sts:AssumeRole
          Path: /
    Policies:
      - PolicyName: LambdaFunctionPolicy
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - logs:CreateLogGroup
                - logs:CreateLogStream
                - logs:PutLogEvents
                - ec2:CreateNetworkInterface
                - ec2:DescribeNetworkInterfaces
                - ec2:DeleteNetworkInterface
              Resource: '*'
  LambdaFunction:
    Type: AWS::Lambda::Function
    Properties:
      Role:
        Fn::GetAtt: LambdaFunctionRole.Arn
      Handler: index.handler
      Code:
        ZipFile: |
          def handler(event, context):
            print("example")
      Runtime: python3.9

[CT.LAMBDA.PR.4] Require an AWS Lambda layer permission to grant access to an AWS organization or specific AWS account

This control checks whether an AWS Lambda layer permission has been configured to grant access to an AWS organization or to a specific AWS account only, by ensuring that public access from all AWS accounts has not been granted to a layer.

- **Control objective:** Enforce least privilege
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Lambda::LayerVersionPermission
• **AWS CloudFormation guard rule:** [CT.LAMBDA.PR.4 rule specification (p. 928)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.LAMBDA.PR.4 rule specification (p. 928)]
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.LAMBDA.PR.4 example templates (p. 930)]

**Explanation**

By default, a layer that you create is **private** to your AWS account. However, you can share the layer with other accounts or make it public, optionally.

A **public** layer may allow unintended access to your source code and applications. A public Lambda layer can expose valuable information about your account, resources, and internal processes.

**Remediation for rule failure**

Set the `OrganizationId` parameter to the ID of an AWS organization, or set the `Principal` parameter to an AWS account ID.

The examples that follow show how to implement this remediation.

**AWS Lambda layer permission - Example one**

An AWS Lambda version permission URL configured to grant layer usage permission to all accounts in an organization. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "LayerVersionPermission": {
        "Type": "AWS::Lambda::LayerVersionPermission",
        "Properties": {
            "Action": "lambda:GetLayerVersion",
            "LayerVersionArn": {
                "Ref": "LayerVersion"
            },
            "OrganizationId": "o-abc123defg"
        }
    }
}
```

**YAML example**

```yaml
LayerVersionPermission:
  Type: AWS::Lambda::LayerVersionPermission
  Properties:
    Action: lambda:GetLayerVersion
    LayerVersionArn: !Ref 'LayerVersion'
    OrganizationId: o-abc123defg
```

The examples that follow show how to implement this remediation.
AWS Control Tower User Guide
Proactive controls

AWS Lambda layer permission - Example two

An AWS Lambda version permission URL configured to grant layer usage permission for an AWS account. The example is shown in JSON and in YAML.

**JSON example**

```
{
   "LayerVersionPermission": {
      "Type": "AWS::Lambda::LayerVersionPermission",
      "Properties": {
         "Action": "lambda:GetLayerVersion",
         "LayerVersionArn": {
            "Ref": "LayerVersion"
         },
         "Principal": "123456789012"
      }
   }
}
```

**YAML example**

```
LayerVersionPermission:
  Type: AWS::Lambda::LayerVersionPermission
  Properties:
    Action: lambda:GetLayerVersion
    LayerVersionArn: !Ref 'LayerVersion'
    Principal: '123456789012'
```

**CT.LAMBDA.PR.4 rule specification**

```
# ####################################################################################################################
##       Rule Specification        
# ####################################################################################################################
#
# Rule Identifier: 
#   lambda_layer_public_access_prohibited_check
#
# Description:
#   This control checks whether an AWS Lambda layer permission has been configured to grant access to an AWS organization or to a specific AWS account only, by ensuring that public access from all AWS accounts has not been granted to a layer.
#
# Reports on: 
#   AWS::Lambda::LayerVersionPermission 
#
# Evaluates: 
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters: 
#   None
#
# Scenarios: 
#   Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
```
And: The input document does not contain any Lambda layer version permission resources
Then: SKIP

Scenario: 2
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a Lambda layer version permission resource
And: 'OrganizationId' has not been provided
And: 'Principal' has been provided and set to '*'
Then: FAIL

Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a Lambda layer version permission resource
And: 'OrganizationId' has not been provided
And: 'Principal' has been provided and set to a non-empty string value other than '*'
Then: PASS

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a Lambda layer version permission resource
And: 'OrganizationId' has not been provided
Then: PASS

Constants

let LAMBDA_LAYER_PERMISSION_TYPE = "AWS::Lambda::LayerVersionPermission"
let INPUT_DOCUMENT = this

Assignments

let lambda_layer_permissions = Resources.*[ Type == %LAMBDA_LAYER_PERMISSION_TYPE ]

Primary Rules

rule lambda_layer_public_access_prohibited_check when is_cfn_template(%INPUT_DOCUMENT)
  %lambda_layer_permissions not empty {
    check(%lambda_layer_permissions.Properties)
    <<
    [CT.LAMBDA.PR.4]: Require an AWS Lambda layer permission to grant access to an AWS organization or specific AWS account
    [FIX]: Set the 'OrganizationId' parameter to the ID of an AWS organization, or set the 'Principal' parameter to an AWS account ID.
    >>
  }
rule lambda_layer_public_access_prohibited_check when is_cfn_hook(%INPUT_DOCUMENT, %LAMBDA_LAYER_PERMISSION_TYPE) {
  check(%INPUT_DOCUMENT.%LAMBDA_LAYER_PERMISSION_TYPE.resourceProperties)
  <<
  [CT.LAMBDA.PR.4]: Require an AWS Lambda layer permission to grant access to an AWS organization or specific AWS account
  [FIX]: Set the 'OrganizationId' parameter to the ID of an AWS organization, or set the 'Principal' parameter to an AWS account ID.
  >>
}

Parameterized Rules

rule check(lambda_layer_permission) {
  %lambda_layer_permission [}
OrganizationId not exists
} {  
  # Scenarios 2 and 3  
  Principal exists  
  check_is_string_and_not_empty(Principal)  
  Principal !="*"
}
%lambda_layer_permission [  
  OrganizationId exists
] {  
  # Scenario 4  
  check_is_string_and_not_empty(OrganizationId)
}

# Utility Rules
#
rule is_cfn_template(doc) {  
  %doc {  
    AWSTemplateFormatVersion exists or  
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {  
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
rule check_is_string_and_not_empty(value) {  
  %value {  
    this is_string  
    this != /\A\s*\z/
  }
}

**CT.LAMBDA.PR.4 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

**Resources:**

LayerVersion:
  
  Type: AWS::Lambda::LayerVersion  
  Properties:
    CompatibleRuntimes:
      - python3.9  
    Content:
      S3Bucket: example-layer-bucket  
      S3Key: layer.zip  
    Description: Example layer  
    LicenseInfo: MIT  
  
LayerVersionPermission:
  
  Type: AWS::Lambda::LayerVersionPermission  
  Properties:
    Action: lambda:GetLayerVersion  
    LayerVersionArn:
      Ref: LayerVersion  
    OrganizationId: o-abc123defg
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  LayerVersion:
    Type: AWS::Lambda::LayerVersion
    Properties:
      CompatibleRuntimes:
        - python3.9
      Content:
        S3Bucket: example-layer-bucket
        S3Key: layer.zip
      Description: Example layer
      LayerName: example-layer
      LicenseInfo: MIT
      LayerVersionPermission:
        Type: AWS::Lambda::LayerVersionPermission
        Properties:
          Action: lambda:GetLayerVersion
          LayerVersionArn:
            Ref: LayerVersion
          Principal: "*"

[CT.LAMBDA.PR.5] Require an AWS Lambda function URL to use AWS IAM-based authentication

This control checks whether an AWS Lambda function URL is configured to use authentication that's based on IAM.

- **Control objective**: Enforce least privilege
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::Lambda::Url
- **AWS CloudFormation guard rule**: [CT.LAMBDA.PR.5 rule specification](p. 932)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.LAMBDA.PR.5 rule specification](p. 932)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.LAMBDA.PR.5 example templates](p. 934)

Explanation

You can control access to a Lambda function URL using the AuthType parameter, combined with resource-based policies that are attached to your specific function. The configuration of these two components determines who can invoke or perform other administrative actions on your function URL.

The AuthType parameter determines how Lambda authenticates or authorizes requests to your Lambda function URL (endpoint). Setting AuthType to NONE means that Lambda does not perform any authentication before it invokes your function. However, your function's resource-based policy is...
always in effect, and the policy must grant public access before your Lambda function URL (endpoint) can receive requests.

Remediation for rule failure

Set the AuthType parameter to AWS_IAM

The examples that follow show how to implement this remediation.

**AWS Lambda function URL - Example**

An AWS Lambda function URL (endpoint) configured with AWS IAM-based authentication. The example is shown in JSON and in YAML.

**JSON example**

```
{
  "FunctionUrl": {
    "Type": "AWS::Lambda::Url",
    "Properties": {
      "TargetFunctionArn": {
        "Fn::GetAtt": [
          "LambdaFunction",
          "Arn"
        ],
        "AuthType": "AWS_IAM"
      }
    }
  }
}
```

**YAML example**

```
FunctionUrl:
  Type: AWS::Lambda::Url
  Properties:
    TargetFunctionArn: !GetAtt 'LambdaFunction.Arn'
    AuthType: AWS_IAM
```

**CT.LAMBDA.PR.5 rule specification**

```
# ##################################################################
##       Rule Specification      ##
# ##################################################################
#
# Rule Identifier:
#   lambda_function_url_auth_check
#
# Description:
#   This control checks whether an AWS Lambda function URL is configured to use
#   authentication that's
#   based on AWS IAM.
#
# Reports on:
#   AWS::Lambda::Url
```
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
#   None
# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document does not contain any Lambda function URL resources
#   Then: SKIP
# Scenario: 2
#   Given: The input document contains a Lambda function URL resource
#   And: 'AuthType' has not been provided
#   Then: FAIL
# Scenario: 3
#   Given: The input document contains a Lambda function URL resource
#   And: 'AuthType' been provided and set to a value other than 'AWS_IAM'
#   Then: FAIL
# Scenario: 4
#   Given: The input document contains a Lambda function URL resource
#   And: 'AuthType' been provided and set to 'AWS_IAM'
#   Then: PASS

# Constants
let LAMBDA_FUNCTION_URL_TYPE = "AWS::Lambda::Url"
let AUTHORIZED_AUTHENTICATION_TYPES = ["AWS_IAM"]
let INPUT_DOCUMENT = this

# Assignments
let lambda_function_urls = Resources.*[ Type == %LAMBDA_FUNCTION_URL_TYPE ]

# Primary Rules
rule lambda_function_url_auth_check when is_cfn_template(%INPUT_DOCUMENT)
  %lambda_function_urls not empty {
    check(%lambda_function_urls.Properties)
    <<
    [CT.LAMBDA.PR.5]: Require an AWS Lambda function URL to use AWS IAM-based authentication
    [FIX]: Set the 'AuthType' parameter to 'AWS_IAM'
    >>
  }
rule lambda_function_url_auth_check when is_cfn_hook(%INPUT_DOCUMENT, %LAMBDA_FUNCTION_URL_TYPE) {
  check(%INPUT_DOCUMENT.%LAMBDA_FUNCTION_URL_TYPE.resourceProperties)
  <<
  [CT.LAMBDA.PR.5]: Require an AWS Lambda function URL to use AWS IAM-based authentication
  [FIX]: Set the 'AuthType' parameter to 'AWS_IAM'
  >>
}
# Parameterized Rules

```
rule check(lambda_function_url) {
  lambda_function_url {
    # Scenario 2
    AuthType exists
    # Scenarios 3 and 4
    AuthType in %AUTHORIZED_AUTHENTICATION_TYPES
  }
}
```

# Utility Rules

```
rule is_cfn_template(doc) {
  doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
```

```
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  doc.%RESOURCE_TYPE.resourceProperties exists
}
```

**CT.LAMBDA.PR.5 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```yaml
Resources:
  LambdaFunctionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service:
              - lambda.amazonaws.com
            Action:
              - sts:AssumeRole
            Path: /
          Policies:
            PolicyName: LambdaFunctionPolicy
            PolicyDocument:
              Version: '2012-10-17'
              Statement:
                - Effect: Allow
                  Action:
                    - logs:CreateLogGroup
                    - logs:CreateLogStream
                    - logs:PutLogEvents
                  Resource: '*'
  LambdaFunction:
    Type: AWS::Lambda::Function
    Properties:
```
Role:
   Fn::GetAtt: LambdaFunctionRole.Arn
Handler: index.handler
Code:
   ZipFile: "def handler(event, context):
       print("example")
   
Runtime: python3.9
FunctionUrl:
   Type: AWS::Lambda::Url
Properties:
   TargetFunctionArn:
      Fn::GetAtt:
         - LambdaFunction
         - Arn
   AuthType: AWS_IAM

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
LambdaFunctionRole:
   Type: AWS::IAM::Role
 Properties:
   AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
         - Effect: Allow
           Principal:
              Service:
                 - lambda.amazonaws.com
           Action:
             - sts:AssumeRole
      Path: /
      Policies:
         - PolicyName: LambdaFunctionPolicy
           PolicyDocument:
              Version: '2012-10-17'
              Statement:
                 - Effect: Allow
                   Action:
                     - logs:CreateLogGroup
                     - logs:CreateLogStream
                     - logs:PutLogEvents
                   Resource: '*'
LambdaFunction:
   Type: AWS::Lambda::Function
 Properties:
   Role:
      Fn::GetAtt: LambdaFunctionRole.Arn
   Handler: index.handler
   Code:
      ZipFile: "def handler(event, context):
          print("example")
      
   Runtime: python3.9
FunctionUrl:
   Type: AWS::Lambda::Url
 Properties:
   TargetFunctionArn:
      Fn::GetAtt:
         - LambdaFunction
         - Arn
   AuthType: NONE

935
[CT.LAMBDA.PR.6] Require an AWS Lambda function URL CORS policy to restrict access to specific origins

This control checks whether an AWS Lambda function URL is configured with a cross-origin resource sharing (CORS) policy that does not grant access to all origins.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Lambda::Url
- **AWS CloudFormation guard rule:** [CT.LAMBDA.PR.6 rule specification (p. 937)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.LAMBDA.PR.6 rule specification (p. 937)]
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.LAMBDA.PR.6 example templates (p. 939)]

Explanation

Cross-Origin Resource Sharing (CORS) is a mechanism based on an HTTP-header, which allows a server to indicate any origins (domain, scheme, or port) other than its own, from which a browser should permit loading resources.

If you set a wildcard origin (*) in a CORS policy, you allow code running in browsers from any origin to gain access to your function URL.

Remediation for rule failure

In the `Cors` parameter, ensure that the value of `AllowOrigins` does not contain wildcard origins (*, http://* and https://*)

The examples that follow show how to implement this remediation.

AWS Lambda Function URL - Example

AWS Lambda function URL configured with a cross-origin resource sharing (CORS) policy that restricts access to a specific origin. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "FunctionUrl": {
        "Type": "AWS::Lambda::Url",
        "Properties": {
            "TargetFunctionArn": {
                "Fn::GetAtt": [
                    "LambdaFunction",
                    "Arn"
                ],
            },
            "AuthType": "AWS_IAM",
            "Cors": {
                "AllowOrigins": [
                    "https://example.com"
                ]
            }
        }
    }
}
```
AWS Control Tower User Guide
Proactive controls

YAML example

```yaml
FunctionUrl:
  Type: AWS::Lambda::Url
Properties:
  TargetFunctionArn: !GetAtt 'LambdaFunction.Arn'
  AuthType: AWS_IAM
  Cors:
    AllowOrigins:
      - https://example.com
```

CT.LAMBDA.PR.6 rule specification

```plaintext
# ######################################################################
## Rule Specification ##
# ######################################################################
#
# Rule Identifier:
#   lambda_function_url_cors_check
#
# Description:
#   This control checks whether an AWS Lambda function URL is configured with a cross-origin resource sharing (CORS) policy that does not grant access to all origins.
#
# Reports on:
#   AWS::Lambda::Url
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation Hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any Lambda function URL resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a Lambda function URL resource
#     And: 'AllowOrigins' in 'Cors' has not been provided or has been provided as an empty list
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a Lambda function URL resource
#     And: 'Cors' has been provided
#     And: 'AllowOrigins' in 'Cors' has been provided as a non-empty list
#     And: 'AllowOrigins' has an entry that contains a wildcard value '*'
#     Then: FAIL
```
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a Lambda function URL resource
# And: 'Cors' has been provided
# And: 'AllowOrigins' in 'Cors' has been provided as a non-empty list
# And: No entries in 'AllowOrigins' contain a wildcard value '*'
# Then: PASS

# Constants
#
let LAMBDA_FUNCTION_URL_TYPE = "AWS::Lambda::Url"
let INPUT_DOCUMENT = this
#
# Assignments
#
let lambda_function_urls = Resources.*[ Type == %LAMBDA_FUNCTION_URL_TYPE ]
#
# Primary Rules
#
rule lambda_function_url_cors_check when is_cfn_template(%INPUT_DOCUMENT)
%lambda_function_urls not empty {
    check(%lambda_function_urls.Properties)
    <<
    [CT.LAMBDA.PR.6]: Require an AWS Lambda function URL CORS policy to restrict access
to specific origins
    [FIX]: In the 'Cors' parameter, ensure that the value of 'AllowOrigins' does not
contain wildcard origins ('*', 'http://*' and 'https://*')
    >>
}
rule lambda_function_url_cors_check when is_cfn_hook(%INPUT_DOCUMENT, %LAMBDA_FUNCTION_URL_TYPE) {
    check(%INPUT_DOCUMENT.%LAMBDA_FUNCTION_URL_TYPE.resourceProperties)
    <<
    [CT.LAMBDA.PR.6]: Require an AWS Lambda function URL CORS policy to restrict access
to specific origins
    [FIX]: In the 'Cors' parameter, ensure that the value of 'AllowOrigins' does not
contain wildcard origins ('*', 'http://*' and 'https://*')
    >>
}
#
# Parameterized Rules
#
rule check(lambda_function_url) {
    %lambda_function_url[
        # Scenario 2
        filter_cors_origins(this)
    ]{
        Cors {
            # Scenarios 3 and 4
            AllowOrigins[*] != /\*/
        }
    }
}
rule filter_cors_origins(lambda_function_url) {
    %lambda_function_url {
        Cors exists
        Cors is_struct
        Cors {
            AllowOrigins exists
        }
    }
}
# Proactive controls

## Utility Rules

```python
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

### CT.LAMBDA.PR.6 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```yaml
Resources:
LambdaFunctionRole:
    Type: AWS::IAM::Role
    Properties:
        AssumeRolePolicyDocument:
            Version: '2012-10-17'
            Statement:
                - Effect: Allow
                  Principal:
                    Service:
                      - lambda.amazonaws.com
                  Action:
                    - sts:AssumeRole
                  Path: /
                  Policies:
                    - PolicyName: LambdaFunctionPolicy
                      PolicyDocument:
                        Version: '2012-10-17'
                        Statement:
                            - Effect: Allow
                              Action:
                                - logs:CreateLogGroup
                                - logs:CreateLogStream
                                - logs:PutLogEvents
                              Resource: '*'

LambdaFunction:
    Type: AWS::Lambda::Function
    Properties:
        Role:
            Fn::GetAtt: LambdaFunctionRole.Arn
        Handler: index.handler
        Code:
            ZipFile: "def handler(event, context):
                  print("example")\n                  "
        Runtime: python3.9
```
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
LambdaFunctionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - lambda.amazonaws.com
          Action:
            - sts:AssumeRole
          Path: /
      Policies:
        - PolicyName: LambdaFunctionPolicy
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Effect: Allow
                Action:
                  - logs:CreateLogGroup
                  - logs:CreateLogStream
                  - logs:PutLogEvents
                Resource: '*'
LambdaFunction:
  Type: AWS::Lambda::Function
  Properties:
    Role:
      Fn::GetAtt: LambdaFunctionRole.Arn
    Handler: index.handler
    Code:
      ZipFile: "def handler(event, context):
        print("example")"
    Runtime: python3.9
FunctionUrl:
  Type: AWS::Lambda::Url
  Properties:
    TargetFunctionArn:
      Fn::GetAtt:
        - LambdaFunction
        - Arn
    AuthType: AWS_IAM
    Cors:
      AllowOrigins:
        - '*'

Amazon Neptune controls

Topics

- [CT.NEPTUNE.PR.1] Require an Amazon Neptune DB cluster to have AWS Identity and Access Management (IAM) database authentication enabled (p. 941)
- [CT.NEPTUNE.PR.2] Require an Amazon Neptune DB cluster to have deletion protection enabled (p. 944)
- [CT.NEPTUNE.PR.3] Require an Amazon Neptune DB cluster to have storage encryption enabled (p. 947)
- [CT.NEPTUNE.PR.4] Require an Amazon Neptune DB cluster to enable Amazon CloudWatch Logs export for audit logs (p. 950)
- [CT.NEPTUNE.PR.5] Require an Amazon Neptune DB cluster to set a backup retention period greater than or equal to seven days (p. 954)

[CT.NEPTUNE.PR.1] Require an Amazon Neptune DB cluster to have AWS Identity and Access Management (IAM) database authentication enabled

This control checks whether an Amazon Neptune cluster has AWS Identity and Access Management (IAM) database authentication enabled.

- **Control objective:** Enforce least privilege, Use strong authentication
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Neptune::DBCluster
- **AWS CloudFormation guard rule:** CT.NEPTUNE.PR.1 rule specification (p. 942)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.NEPTUNE.PR.1 rule specification (p. 942)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.NEPTUNE.PR.1 example templates (p. 944)

Explanation

You can use AWS Identity and Access Management (IAM) to authenticate to your Neptune DB instance or DB cluster. IAM allows you to manage access to your database resources centrally, instead of managing access individually on each DB instance or cluster.

Remediation for rule failure

Set the value of the IamAuthEnabled parameter to true.

The examples that follow show how to implement this remediation.

Amazon Neptune Cluster - Example

Neptune Cluster configured with AWS IAM database authentication enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{}
```
"NeptuneDBCluster": {  
    "Type": "AWS::Neptune::DBCluster",  
    "Properties": {  
        "IamAuthEnabled": true  
    }  
}

YAML example

NeptuneDBCluster:  
  Type: AWS::Neptune::DBCluster  
  Properties:  
    IamAuthEnabled: true

CT.NEPTUNE.PR.1 rule specification

# ##############################################################################  
##       Rule Specification       ##  
# ##############################################################################
#  
# Rule Identifier:  
#   neptune_cluster_iam_database_authentication_check  
#  
# Description:  
#   This control checks whether an Amazon Neptune cluster has AWS Identity and Access Management (IAM) database authentication enabled.  
#  
# Reports on:  
#   AWS::Neptune::DBCluster  
#  
# Evaluates:  
#   AWS CloudFormation, AWS CloudFormation hook  
#  
# Rule Parameters:  
#   None  
#  
# Scenarios:  
#  
# Scenario: 1  
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document  
#   And: The input document does not contain any Neptune DB cluster resources  
#   Then: SKIP  
#  
# Scenario: 2  
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document  
#   And: The input document contains a Neptune DB cluster resource  
#   And: 'IamAuthEnabled' has not been provided  
#   Then: FAIL  
#  
# Scenario: 3  
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document  
#   And: The input document contains a Neptune DB cluster resource  
#   And: 'IamAuthEnabled' has been provided and set to a value other than bool(true)  
#   Then: FAIL  
#  
# Scenario: 4  
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a Neptune DB cluster resource  
And: 'IamAuthEnabled' has been provided and set to bool(true)  
Then: PASS

# Constants  
let NEPTUNE_CLUSTER_TYPE = "AWS::Neptune::DBCluster"  
let INPUT_DOCUMENT = this

# Assignments  
let neptune_db_clusters = Resources.*[ Type == %NEPTUNE_CLUSTER_TYPE ]

# Primary Rules  
rule neptune_cluster_iam_database_authentication_check when 
is_cfn_template(%INPUT_DOCUMENT)  
%neptune_db_clusters not empty
{
  check(%neptune_db_clusters.Properties)
  <<
    [CT.NEPTUNE.PR.1]: Require an Amazon Neptune DB cluster to have AWS Identity and  
    Access Management (IAM) database authentication enabled
    [FIX]: Set the value of the 'IamAuthEnabled' parameter to true.
  >>
}

rule neptune_cluster_iam_database_authentication_check when is_cfn_hook(%INPUT_DOCUMENT, 
%NEPTUNE_CLUSTER_TYPE) {
  check(%INPUT_DOCUMENT.%NEPTUNE_CLUSTER_TYPE.resourceProperties)
  <<
    [CT.NEPTUNE.PR.1]: Require an Amazon Neptune DB cluster to have AWS Identity and  
    Access Management (IAM) database authentication enabled
    [FIX]: Set the value of the 'IamAuthEnabled' parameter to true.
  >>
}

# Parameterized Rules  
# rule check(neptune_cluster) {
%neptune_cluster {
  # Scenario 2  
  IamAuthEnabled exists  
  # Scenarios 3 and 4  
  IamAuthEnabled == true
}

# Utility Rules  
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or  
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.NEPTUNE.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
NeptuneDBCluster:
  Type: AWS::Neptune::DBCluster
  Properties:
    IamAuthEnabled: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
NeptuneDBCluster:
  Type: AWS::Neptune::DBCluster
  Properties:
    IamAuthEnabled: false

[CT.NEPTUNE.PR.2] Require an Amazon Neptune DB cluster to have deletion protection enabled

This control checks whether an Amazon Neptune cluster has deletion protection enabled.

- Control objective: Improve availability, Protect configurations
- Implementation: AWS CloudFormation guard rule
- Control behavior: Proactive
- Resource types: AWS::Neptune::DBCluster
- AWS CloudFormation guard rule: CT.NEPTUNE.PR.2 rule specification (p. 945)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.NEPTUNE.PR.2 rule specification (p. 945)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.NEPTUNE.PR.2 example templates (p. 947)

Explanation

Cluster deletion protection adds an additional layer of protection against accidental database deletion, or deletion by an unauthorized entity. A Neptune cluster cannot be deleted while deletion protection is enabled. Deletion protection must be disabled first, before a delete request can succeed.

Remediation for rule failure

Set the value of the DeletionProtection parameter to true.

The examples that follow show how to implement this remediation.
Amazon Neptune Cluster - Example

An Amazon Neptune Cluster configured with deletion protection enabled. The example is shown in JSON and in YAML.

JSON example

```json
{
  "NeptuneDBCluster": {
    "Type": "AWS::Neptune::DBCluster",
    "Properties": {
      "DeletionProtection": true
    }
  }
}
```

YAML example

```yaml
NeptuneDBCluster:
  Type: AWS::Neptune::DBCluster
  Properties:
    DeletionProtection: true
```

CT.NEPTUNE.PR.2 rule specification

```text
# ###################################
##       Rule Specification        
###################################
#
# Rule Identifier:
#   neptune_cluster_deletion_protection_enabled_check
#
# Description:
#   This control checks whether an Amazon Neptune cluster has deletion protection enabled.
#
# Reports on:
#   AWS::Neptune::DBCluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any Neptune DB cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a Neptune DB cluster resource
#     And: 'DeletionProtection' has not been provided
#     Then: FAIL
```
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a Neptune DB cluster resource
#   And: 'DeletionProtection' has been provided and set to a value other than
#        bool(true)
# Then: FAIL

# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a Neptune DB cluster resource
#   And: 'DeletionProtection' has been provided and set to bool(true)
# Then: PASS

# Constants
#
let NEPTUNE_CLUSTER_TYPE = "AWS::Neptune::DBCluster"
let INPUT_DOCUMENT = this
#
# Assignments
#
let neptune_db_clusters = Resources.*[ Type == %NEPTUNE_CLUSTER_TYPE ]
#
# Primary Rules
#
rule neptune_cluster_deletion_protection_enabled_check when
  is_cfn_template(%INPUT_DOCUMENT)
  %neptune_db_clusters not empty
  { check(%neptune_db_clusters.Properties)
    <<
    [CT.NEPTUNE.PR.2]: Require an Amazon Neptune DB cluster to have deletion protection
    enabled
    [FIX]: Set the value of the 'DeletionProtection' parameter to true.
    >>
  }
rule neptune_cluster_deletion_protection_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %NEPTUNE_CLUSTER_TYPE) {
  check(%INPUT_DOCUMENT.%NEPTUNE_CLUSTER_TYPE.resourceProperties)
  <<
  [CT.NEPTUNE.PR.2]: Require an Amazon Neptune DB cluster to have deletion protection
  enabled
  [FIX]: Set the value of the 'DeletionProtection' parameter to true.
  >>
}
#
# Parameterized Rules
#
rule check(neptune_cluster) {
  %neptune_cluster {
  # Scenario 2
  DeletionProtection exists
  # Scenarios 3 and 4
  DeletionProtection == true
  }
}
#
# Utility Rules
#
rule is_cfn_template(doc) {
}
You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  NeptuneDBCluster:
    Type: AWS::Neptune::DBCluster
    Properties:
      DeletionProtection: true
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
  NeptuneDBCluster:
    Type: AWS::Neptune::DBCluster
    Properties:
      DeletionProtection: false
```

[CT.NEPTUNE.PR.3] Require an Amazon Neptune DB cluster to have storage encryption enabled

This control checks whether an Amazon Neptune cluster has storage encryption enabled.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Neptune::DBCluster
- **AWS CloudFormation guard rule:** CT.NEPTUNE.PR.3 rule specification (p. 948)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.NEPTUNE.PR.3 rule specification (p. 948)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.NEPTUNE.PR.3 example templates (p. 950)

Explanation
Neptune encrypted instances provide an additional layer of data protection, because they help to secure your data from unauthorized access to the underlying storage. Neptune encryption helps increase data protection of your applications that are deployed in the cloud. You also can use it to fulfill compliance requirements for data-at-rest encryption.

**Usage considerations**

- This control checks only whether the `StorageEncrypted` property is provided and set to `true`. When you create an encrypted Neptune DB instance, you also can supply the AWS KMS key identifier for your encryption key by means of the `KmsKeyId` property. If you don’t specify an AWS KMS key identifier, Neptune uses your default Amazon RDS encryption key (aws/rds) for your new Neptune DB instance.

**Remediation for rule failure**

Set `StorageEncrypted` to `true`.

The examples that follow show how to implement this remediation.

**Amazon Neptune Cluster - Example**

An Amazon Neptune Cluster configured with storage encryption enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "NeptuneDBCluster": {
      "Type": "AWS::Neptune::DBCluster",
      "Properties": {
         "StorageEncrypted": true
      }
   }
}
```

**YAML example**

```yaml
NeptuneDBCluster:
   Type: AWS::Neptune::DBCluster
   Properties:
      StorageEncrypted: true
```

**CT.NEPTUNE.PR.3 rule specification**

```bash
# # Rule Specification #
# # Rule Identifier:
# #   neptune_cluster_encrypted_check
# # Description:
# #   This control checks whether an Amazon Neptune cluster has storage encryption enabled.
```
# Reports on:
# AWS::Neptune::DBCluster
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
#  Scenario: 1
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#        And: The input document does not contain any Neptune DB cluster resources
#        Then: SKIP
#  Scenario: 2
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#        And: The input document contains a Neptune DB cluster resource
#        And: 'StorageEncrypted' has not been provided
#        Then: FAIL
#  Scenario: 3
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#        And: The input document contains a Neptune DB cluster resource
#        And: 'StorageEncrypted' has been provided and set to a value other than bool(true)
#        Then: FAIL
#  Scenario: 4
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#        And: The input document contains a Neptune DB cluster resource
#        And: 'StorageEncrypted' has been provided and set to bool(true)
#        Then: PASS
#
# Constants
#
let NEPTUNE_CLUSTER_TYPE = "AWS::Neptune::DBCluster"
let INPUT_DOCUMENT = this
#
# Assignments
#
let neptune_db_clusters = Resources.*[ Type == %NEPTUNE_CLUSTER_TYPE ]
#
# Primary Rules
#
rule neptune_cluster_encrypted_check when is_cfn_template(%INPUT_DOCUMENT)
  %neptune_db_clusters not empty {
    check(%neptune_db_clusters.Properties)
    <<
      [CT.NEPTUNE.PR.3]: Require an Amazon Neptune DB cluster to have storage encryption enabled
      [FIX]: Set 'StorageEncrypted' to 'true'.
    >>}

rule neptune_cluster_encrypted_check when is_cfn_hook(%INPUT_DOCUMENT, %NEPTUNE_CLUSTER_TYPE) {
  check(%INPUT_DOCUMENT.%NEPTUNE_CLUSTER_TYPE.resourceProperties)
  <<
    [CT.NEPTUNE.PR.3]: Require an Amazon Neptune DB cluster to have storage encryption enabled
    [FIX]: Set 'StorageEncrypted' to 'true'.
  >>}
CT.NEPTUNE.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  NeptuneDBCluster:
    Type: AWS::Neptune::DBCluster
    Properties:
      StorageEncrypted: true
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
  NeptuneDBCluster:
    Type: AWS::Neptune::DBCluster
    Properties:
      StorageEncrypted: false
```

[CT.NEPTUNE.PR.4] Require an Amazon Neptune DB cluster to enable Amazon CloudWatch Logs export for audit logs

This control checks whether an Amazon Neptune cluster is configured to send audit logs to Amazon CloudWatch Logs.
• **Control objective:** Establish logging and monitoring
• **Implementation:** AWS CloudFormation guard rule
• **Control behavior:** Proactive
• **Resource types:** AWS::Neptune::DBCluster
• **AWS CloudFormation guard rule:** [CT.NEPTUNE.PR.4 rule specification (p. 952)](p. 952)

**Details and examples**

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.NEPTUNE.PR.4 rule specification (p. 952)](p. 952)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.NEPTUNE.PR.4 example templates (p. 953)](p. 953)

**Explanation**

You can configure an Amazon Neptune DB cluster to publish audit log data to a log group in Amazon CloudWatch Logs. Storing your Neptune DB cluster audit log data in Amazon CloudWatch Logs allows you to perform real-time analysis of the log data, and also to use Amazon CloudWatch to create alarms and view metrics.

**Remediation for rule failure**

In the `EnableCloudwatchLogsExports` parameter, set an entry to the value `audit`.

The examples that follow show how to implement this remediation.

**Amazon Neptune cluster - Example**

An Amazon Neptune Cluster configured to export audit logs to Amazon CloudWatch Logs. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "NeptuneDBCluster": {
    "Type": "AWS::Neptune::DBCluster",
    "Properties": {
      "EnableCloudwatchLogsExports": [
        "audit"
      ]
  }
}
```

**YAML example**

```yaml
NeptuneDBCluster:
  Type: AWS::Neptune::DBCluster
  Properties:
    EnableCloudwatchLogsExports:
      - audit
```
CT.NEPTUNE.PR.4 rule specification

```plaintext
# # ###################################
##       Rule Specification        
# ###################################
#
# Rule Identifier:
#   neptune_cluster_cloudwatch_audit_log_export_enabled
#
# Description:
#   This control checks whether an Amazon Neptune cluster is configured to send audit logs
to Amazon CloudWatch Logs.
#
# Reports on:
#   AWS::Neptune::DBCluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any Neptune DB cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document contains a Neptune DB cluster resource
#     And: 'EnableCloudwatchLogsExports' has not been provided or has been provided as an
#          empty list
#     Then: FAIL
#   Scenario: 3
#     Given: The input document contains a Neptune DB cluster resource
#     And: 'EnableCloudwatchLogsExports' has been provided as a non-empty list
#     And: 'EnableCloudwatchLogsExports' does not contain an entry with the value 'audit'
#     Then: FAIL
#   Scenario: 4
#     Given: The input document contains a Neptune DB cluster resource
#     And: 'EnableCloudwatchLogsExports' has been provided as a non-empty list
#     And: 'EnableCloudwatchLogsExports' contains an entry with the value 'audit'
#     Then: PASS
#
# Constants
#
let NEPTUNE_CLUSTER_TYPE = "AWS::Neptune::DBCluster"
let INPUT_DOCUMENT = this
#
# Assignments
#
let neptune_db_clusters = Resources.*[ Type == %NEPTUNE_CLUSTER_TYPE ]
#
# Primary Rules
#
```

952
CT.NEPTUNE.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:

NeptuneDBCluster:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```yaml
Resources:
  NeptuneDBCluster:
    Type: AWS::Neptune::DBCluster
    Properties:
      EnableCloudwatchLogsExports:
        - slowquery
```

[CT.NEPTUNE.PR.5] Require an Amazon Neptune DB cluster to set a backup retention period greater than or equal to seven days

This control checks whether Amazon Neptune DB clusters have configured automatic backups with a retention period set to 7 or more days (>=7). The default retention period is one day.

- **Control objective**: Improve resiliency
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::Neptune::DBCluster
- **AWS CloudFormation guard rule**: CT.NEPTUNE.PR.5 rule specification (p. 955)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.NEPTUNE.PR.5 rule specification (p. 955)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.NEPTUNE.PR.5 example templates (p. 957)

Explanation

Amazon Neptune backs up your cluster volume automatically, and it retains restore data for the length of the backup retention period. Backups are continuous and incremental. You can restore to any point within the backup retention period, quickly. No performance impact or interruption of database service occurs as backup data is being written.

Remediation for rule failure

Set the BackupRetentionPeriod parameter to an integer value between 7 and 35 days (inclusive).

The examples that follow show how to implement this remediation.

Amazon Neptune cluster - Example

An Amazon Neptune Cluster configured with a backup retention period of seven (7) days. The example is shown in JSON and in YAML.
## JSON example

```json
{
    "NeptuneDBCluster": {
        "Type": "AWS::Neptune::DBCluster",
        "Properties": {
            "BackupRetentionPeriod": 7
        }
    }
}
```

## YAML example

```yaml
NeptuneDBCluster:
  Type: AWS::Neptune::DBCluster
  Properties:
    BackupRetentionPeriod: 7
```

## CT.NEPTUNE.PR.5 rule specification

```
# ###################################
##       Rule Specification        ##
#####################################
#
# Rule Identifier:
#   neptune_cluster_backup_retention_check
#
# Description:
#   This control checks whether Amazon Neptune DB clusters have configured automatic
#   backups with a retention period set to 7 or more days (>=7). The default retention period
#   is one day.
#
# Reports on:
#   AWS::Neptune::DBCluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any Neptune DB cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a Neptune DB cluster resource
#     And: 'BackupRetentionPeriod' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
```
And: The input document contains a Neptune DB cluster resource
And: 'BackupRetentionPeriod' has been provided and set to an integer value
less than seven (< 7)
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document contains a Neptune DB cluster resource
And: 'BackupRetentionPeriod' has been provided and set to an integer value
greater than or equal to seven (>=7)
Then: PASS

Constants

let NEPTUNE_CLUSTER_TYPE = "AWS::Neptune::DBCluster"
let MINIMUM_BACKUP_RETENTION_PERIOD = 7
let INPUT_DOCUMENT = this

Assignments

let neptune_db_clusters = Resources.*[ Type == %NEPTUNE_CLUSTER_TYPE ]

Primary Rules

rule neptune_cluster_backup_retention_check when is_cfn_template(%INPUT_DOCUMENT)
%neptune_db_clusters not empty {
  check(%neptune_db_clusters.Properties)
  <<
  [CT.NEPTUNE.PR.5]: Require an Amazon Neptune DB cluster to set a backup retention
  period greater than or equal to seven days
  [FIX]: Set the 'BackupRetentionPeriod' parameter to an integer value between 7 and
  35 days (inclusive).
  >>
}

rule neptune_cluster_backup_retention_check when is_cfn_hook(%INPUT_DOCUMENT, %NEPTUNE_CLUSTER_TYPE) {
  check(%INPUT_DOCUMENT.%NEPTUNE_CLUSTER_TYPE.resourceProperties)
  <<
  [CT.NEPTUNE.PR.5]: Require an Amazon Neptune DB cluster to set a backup retention
  period greater than or equal to seven days
  [FIX]: Set the 'BackupRetentionPeriod' parameter to an integer value between 7 and
  35 days (inclusive).
  >>
}

Parameterized Rules

rule check(neptune_cluster) {
  %neptune_cluster {
    # Scenario 2
    BackupRetentionPeriod exists
    # Scenarios 3 and 4
    BackupRetentionPeriod >= %MINIMUM_BACKUP_RETENTION_PERIOD
  }
}

Utility Rules

rule is_cfn_template(doc) {
  %doc {
AWS Control Tower User Guide
Proactive controls

CT.NEPTUNE.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  NeptuneDBCluster:
    Type: AWS::Neptune::DBCluster
    Properties:
      BackupRetentionPeriod: 7

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  NeptuneDBCluster:
    Type: AWS::Neptune::DBCluster
    Properties:
      BackupRetentionPeriod: 1

AWS Network Firewall controls

Topics

- [CT.NETWORK-FIREWALL.PR.1] Require any AWS Network Firewall firewall policy to have an associated rule group (p. 957)
- [CT.NETWORK-FIREWALL.PR.2] Require any AWS Network Firewall firewall policy to drop or forward stateless full packets by default when they do not match a rule (p. 962)
- [CT.NETWORK-FIREWALL.PR.3] Require any AWS Network Firewall firewall policy to drop or forward fragmented packets by default when they do not match a stateless rule (p. 967)
- [CT.NETWORK-FIREWALL.PR.4] Require any AWS Network Firewall rule group to contain at least one rule (p. 972)

[CT.NETWORK-FIREWALL.PR.1] Require any AWS Network Firewall firewall policy to have an associated rule group

This control checks whether there is at least one stateful or stateless rule group associated with an AWS Network Firewall firewall policy.

- **Control objective:** Limit network access
**Implementation**: AWS CloudFormation Guard Rule

**Control behavior**: Proactive

**Resource types**: AWS::NetworkFirewall::FirewallPolicy


### Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.NETWORK-FIREWALL.PR.1 rule specification (p. 959)](https://docs.aws.amazon.com/network-firewall/latest/userguide/ct-network-firewall-pr.1.html)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.NETWORK-FIREWALL.PR.1 example templates (p. 961)](https://docs.aws.amazon.com/network-firewall/latest/userguide/ct-network-firewall-pr.1.html)

### Explanation

A firewall policy defines how your firewall monitors and handles traffic in Amazon Virtual Private Cloud (Amazon VPC). Configuration of stateless and stateful rule groups helps to filter packets and traffic flows, and to define the default traffic handling settings.

**Remediation for rule failure**

Within the FirewallPolicy definition, refer to one or more rule groups in StatefulRuleGroupReferences or StatelessRuleGroupReferences.

The examples that follow show how to implement this remediation.

**AWS Network Firewall Firewall Policy - Example**

AWS Network Firewall firewall policy configured with stateful and stateless rule group associations. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "FirewallPolicy": {
        "Type": "AWS::NetworkFirewall::FirewallPolicy",
        "Properties": {
            "FirewallPolicyName": "sample-firewall-policy",
            "FirewallPolicy": {
                "StatelessDefaultActions": [
                    "aws:forward_to_sfe"
                ],
                "StatelessFragmentDefaultActions": [
                    "aws:drop"
                ],
                "StatefulRuleGroupReferences": [
                    {
                        "ResourceArn": {
                            "Ref": "StatefulRuleGroup"
                        }
                    }
                ],
                "StatelessRuleGroupReferences": [
                    {
                        "ResourceArn": {
                            "Ref": "StatelessRuleGroup"
                        },
                        "Priority": 100
                    }
                ]
            }
        }
    }
}
```

958
YAML example

FirewallPolicy:
  Type: AWS::NetworkFirewall::FirewallPolicy
  Properties:
    FirewallPolicyName: sample-firewall-policy
    FirewallPolicy:
      StatelessDefaultActions:
        - aws:forward_to_sfe
      StatelessFragmentDefaultActions:
        - aws:drop
      StatefulRuleGroupReferences:
        - ResourceArn: !Ref 'StatefulRuleGroup'
      StatelessRuleGroupReferences:
        - ResourceArn: !Ref 'StatelessRuleGroup'
      Priority: 100

CT.NETWORK-FIREWALL.PR.1 rule specification

```yaml
# ###################################################################
##       Rule Specification       ##
###################################################################
#
# Rule Identifier:
#   netfw_policy_rule_group_associated_check
#
# Description:
#   This control checks whether there is at least one stateful or stateless rule group
#   associated with an AWS Network Firewall firewall policy.
#
# Reports on:
#   AWS::NetworkFirewall::FirewallPolicy
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     And: The input document does not contain any NetworkFirewall firewall policy
#          resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     And: 'StatefulRuleGroupReferences' has not been provided in 'FirewallPolicy'
#```
# And: 'StatelessRuleGroupReferences' has not been provided in 'FirewallPolicy'
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a NetworkFirewall firewall policy resource
# And: 'StatefulRuleGroupReferences' has not been provided in 'FirewallPolicy'
# And: 'StatelessRuleGroupReferences' has been provided as an empty list
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a NetworkFirewall firewall policy resource
# And: 'StatelessRuleGroupReferences' has not been provided in 'FirewallPolicy'
# And: 'StatefulRuleGroupReferences' has been provided as an empty list
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a NetworkFirewall firewall policy resource
# And: 'StatelessRuleGroupReferences' has been provided as an empty list
# And: 'StatefulRuleGroupReferences' has been provided as an empty list
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a NetworkFirewall firewall policy resource
# And: One or both of 'StatelessRuleGroupReferences' and 'StatefulRuleGroupReferences' have been provided as a non-empty list
# Then: PASS

# Constants

let NETFW_FIREWALL_POLICY_TYPE = "AWS::NetworkFirewall::FirewallPolicy"
let INPUT_DOCUMENT = this

# Assignments

let netfw_firewall_policies = Resources.*[ Type == %NETFW_FIREWALL_POLICY_TYPE ]

# Primary Rules

rule netfw_policy_rule_group_associated_check when is_cfn_template(%INPUT_DOCUMENT)

  %netfw_firewall_policies not empty {

    check(%netfw_firewall_policies.Properties)
    <<
    [CT.NETWORK-FIREWALL.PR.1]: Require any AWS Network Firewall firewall policy to have an associated rule group
    [FIX]: Within the 'FirewallPolicy' definition, refer to one or more rule groups in 'StatefulRuleGroupReferences' or 'StatelessRuleGroupReferences'.
    >>

  }
rule netfw_policy_rule_group_associated_check when is_cfn_hook(%INPUT_DOCUMENT,

  %NETFW_FIREWALL_POLICY_TYPE) {

  check(%INPUT_DOCUMENT.%NETFW_FIREWALL_POLICY_TYPE.resourceProperties)
  <<
  [CT.NETWORK-FIREWALL.PR.1]: Require any AWS Network Firewall firewall policy to have an associated rule group
  [FIX]: Within the 'FirewallPolicy' definition, refer to one or more rule groups in 'StatefulRuleGroupReferences' or 'StatelessRuleGroupReferences'.
  >>

CT.NETWORK-FIREWALL.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
StatefulRuleGroup:
  Type: AWS::NetworkFirewall::RuleGroup
  Properties:
    RuleGroupName:
      Fn::Sub: ${AWS::StackName}-stateful-example
    Type: STATEFUL
    RuleGroup:
      RulesString: pass tcp 10.20.20.0/24 45400:45500 < 10.10.10.0/24 5203
      (msg:"test";sid:1;rev:1;)
    Capacity: 100
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

[CT.NETWORK-FIREWALL.PR.2] Require any AWS Network Firewall firewall policy to drop or forward stateless full packets by default when they do not match a rule

This control checks whether an AWS Network Firewall firewall policy is configured with a user-defined stateless default action for full packets.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::NetworkFirewall::FirewallPolicy
- **AWS CloudFormation guard rule:** [CT.NETWORK-FIREWALL.PR.2 rule specification (p. 964)]
Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.NETWORK-FIREWALL.PR.2 rule specification](p. 964)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.NETWORK-FIREWALL.PR.2 example templates](p. 966)

Explanation

A firewall policy defines how your firewall monitors and handles traffic in Amazon VPC. You configure stateless and stateful rule groups to filter packets and traffic flows. Defaulting to Pass can allow unintended traffic.

Remediation for rule failure

Within `FirewallPolicy`, include one of `aws:drop` or `aws:forward_to_sfe` in `StatelessDefaultActions`.

The examples that follow show how to implement this remediation.

**AWS Network Firewall Firewall Policy - Example One**

AWS Network Firewall firewall policy configured with a stateless default action to drop full packets. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "FirewallPolicy": {
      "Type": "AWS::NetworkFirewall::FirewallPolicy",
      "Properties": {
         "FirewallPolicyName": {
            "Fn::Sub": "${AWS::StackName}-sample"
         },
         "FirewallPolicy": {
            "StatelessFragmentDefaultActions": ["aws:forward_to_sfe"],
            "StatelessDefaultActions": ["aws:drop"]
         }
      }
   }
}
```

**YAML example**

```
FirewallPolicy:
   Type: AWS::NetworkFirewall::FirewallPolicy
Properties:
   FirewallPolicyName: !Sub '${AWS::StackName}-sample'
   FirewallPolicy:
     StatelessFragmentDefaultActions:
       - aws:forward_to_sfe
     StatelessDefaultActions:
       - aws:drop
```

963
The examples that follow show how to implement this remediation.

**AWS Network Firewall Firewall Policy - Example Two**

AWS Network Firewall firewall policy configured with a stateless default action to forward full packets to the stateful rule engine for further inspection. The example is shown in JSON and in YAML.

**JSON example**

```
{
    "FirewallPolicy": {
        "Type": "AWS::NetworkFirewall::FirewallPolicy",
        "Properties": {
            "FirewallPolicyName": {
                "Fn::Sub": "${AWS::StackName}-sample"
            },
            "FirewallPolicy": {
                "StatelessFragmentDefaultActions": [
                    "aws:forward_to_sfe"
                ],
                "StatelessDefaultActions": [
                    "aws:forward_to_sfe"
                ]
            }
        }
    }
}
```

**YAML example**

```
FirewallPolicy:
  Type: AWS::NetworkFirewall::FirewallPolicy
  Properties:
    FirewallPolicyName: !Sub '${AWS::StackName}-sample'
    FirewallPolicy:
      StatelessFragmentDefaultActions:
        - aws:forward_to_sfe
      StatelessDefaultActions:
        - aws:forward_to_sfe
```

**CT.NETWORK-FIREWALL.PR.2 rule specification**

```
# ###################################
##       Rule Specification        
####################################
#
# Rule Identifier:
#   netfw_policy_default_action_full_packets_check
#
# Description:
#   This control checks whether an AWS Network Firewall firewall policy is configured with a user-defined stateless default action for full packets.
#
# Reports on:
```
AWS Control Tower User Guide
Proactive controls

# AWS::NetworkFirewall::FirewallPolicy

# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document does not contain any Network Firewall firewall policy
resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a Network Firewall firewall policy resource
# And: 'StatelessDefaultActions' has not been provided in 'FirewallPolicy'
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a Network Firewall firewall policy resource
# And: 'StatelessDefaultActions' has been provided in 'FirewallPolicy' as an empty
list
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a Network Firewall firewall policy resource
# And: 'StatelessDefaultActions' has been provided in 'FirewallPolicy' as a list that
does not contain
# one of 'aws:drop' or 'aws:forward_to_sfe'
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a Network Firewall firewall policy resource
# And: 'StatelessDefaultActions' has been provided in 'FirewallPolicy' as a list that
contains either
# 'aws:drop' or 'aws:forward_to_sfe'
# Then: PASS

# Constants

let NETFW_FIREWALL_POLICY_TYPE = "AWS::NetworkFirewall::FirewallPolicy"
let INPUT_DOCUMENT = this
let ALLOWED_STATELESS_ACTIONS_LIST = [ "aws:drop", "aws:forward_to_sfe" ]

# Assignments

let netfw_firewall_policies = Resources.*[ Type == %NETFW_FIREWALL_POLICY_TYPE ]

# Primary Rules

rule netfw_policy_default_action_full_packets_check when is_cfn_template(%INPUT_DOCUMENT)
empty {
    check(%netfw_firewall_policies.Properties)
    <<
[CT.NETWORK-FIREWALL.PR.2]: Require any AWS Network Firewall firewall policy to drop or forward stateless full packets by default when they do not match a rule

[FIX]: Within 'FirewallPolicy', include one of 'aws:drop' or 'aws:forward_to_sfe' in 'StatelessDefaultActions'.

>{{

```
rule netfw_policy_default_action_full_packets_check when is_cfn_hook(%INPUT_DOCUMENT, %NETFW_FIREWALL_POLICY_TYPE) {
  check(%INPUT_DOCUMENT.%NETFW_FIREWALL_POLICY_TYPE.resourceProperties)

  [CT.NETWORK-FIREWALL.PR.2]: Require any AWS Network Firewall firewall policy to drop or forward stateless full packets by default when they do not match a rule
  [FIX]: Within 'FirewallPolicy', include one of 'aws:drop' or 'aws:forward_to_sfe' in 'StatelessDefaultActions'.

  >>
}
```

# Parameterized Rules

```# Parameterized Rules
```

```#
```

```rule check(netfw_firewall_policy) {
  %netfw_firewall_policy {
    # Scenario 2
    FirewallPolicy exists
    FirewallPolicy is_struct

    FirewallPolicy {
      StatelessDefaultActions exists
      # Scenario 3
      StatelessDefaultActions is_list
      StatelessDefaultActions not empty

      # Scenario 4 and 5
      some StatelessDefaultActions[*] {
        this in %ALLOWED_STATELESS_ACTIONS_LIST
      }
    }
  }
}
```

# Utility Rules

```# Utility Rules
```

```rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
```

CT.NETWORK-FIREWALL.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
FirewallPolicy:
  Type: AWS::NetworkFirewall::FirewallPolicy
  Properties:
    FirewallPolicyName:
      Fn::Sub: ${AWS::StackName}-example
    FirewallPolicy:
      StatelessFragmentDefaultActions:
        - aws:forward_to_sfe
      StatelessDefaultActions:
        - aws:drop

[CT.NETWORK-FIREWALL.PR.3] Require any AWS Network Firewall firewall policy to drop or forward fragmented packets by default when they do not match a stateless rule

This control checks whether an AWS Network Firewall firewall policy is configured with a default action to drop or forward fragmented packets, when the packets do not match a stateless rule.

- **Control objective**: Limit network access
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::NetworkFirewall::FirewallPolicy
- **AWS CloudFormation guard rule**: [CT.NETWORK-FIREWALL.PR.3 rule specification (p. 969)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.NETWORK-FIREWALL.PR.3 rule specification (p. 969)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.NETWORK-FIREWALL.PR.3 example templates (p. 971)]

Explanation

A firewall policy defines how your firewall monitors and handles traffic in Amazon VPC. You configure stateless and stateful rule groups to filter packets and traffic flows. Defaulting to Pass can allow unintended traffic.
Remediation for rule failure

Within FirewallPolicy, include one of `aws:drop` or `aws:forward_to_sfe` in `StatelessFragmentDefaultActions`.

The examples that follow show how to implement this remediation.

**AWS Network Firewall Firewall Policy - Example One**

AWS Network Firewall firewall policy configured with a stateless default action to drop fragmented packets. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "FirewallPolicy": {
        "Type": "AWS::NetworkFirewall::FirewallPolicy",
        "Properties": {
            "FirewallPolicyName": {
                "Fn::Sub": "${AWS::StackName}-sample"
            },
            "FirewallPolicy": {
                "StatelessDefaultActions": [
                    "aws:forward_to_sfe"
                ],
                "StatelessFragmentDefaultActions": [
                    "aws:drop"
                ]
            }
        }
    }
}
```

**YAML example**

```yaml
FirewallPolicy:
  Type: AWS::NetworkFirewall::FirewallPolicy
Properties:
  FirewallPolicyName: !Sub '${AWS::StackName}-sample'
  FirewallPolicy:
    StatelessDefaultActions:
      - aws:forward_to_sfe
    StatelessFragmentDefaultActions:
      - aws:drop
```

The examples that follow show how to implement this remediation.

**AWS Network Firewall Firewall Policy - Example Two**

AWS Network Firewall firewall policy configured with a stateless default action to forward fragmented packets to the stateful rule engine for further inspection. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "FirewallPolicy": {
```
"Type": "AWS::NetworkFirewall::FirewallPolicy",
"Properties": {
  "FirewallPolicyName": {
    "Fn::Sub": "${AWS::StackName}-sample"
  },
  "FirewallPolicy": {
    "StatelessDefaultActions": [
      "aws:forward_to_sfe"
    ],
    "StatelessFragmentDefaultActions": [
      "aws:forward_to_sfe"
    ]
  }
}

YAML example

FirewallPolicy:
  Type: AWS::NetworkFirewall::FirewallPolicy
  Properties:
    FirewallPolicyName: !Sub '${AWS::StackName}-sample'
    FirewallPolicy:
      StatelessDefaultActions:
        - aws:forward_to_sfe
      StatelessFragmentDefaultActions:
        - aws:forward_to_sfe

CT.NETWORK-FIREWALL.PR.3 rule specification

```plaintext
# ********************
##   Rule Specification   ##
# ********************
#
# Rule Identifier:
#   netfw_policy_default_action_fragment_packets_check
#
# Description:
#   This control checks whether an AWS Network Firewall firewall policy is configured with
#   a default action to drop or forward fragmented packets, when the packets do not match a
#   stateless rule.
#
# Reports on:
#   AWS::NetworkFirewall::FirewallPolicy
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#     And: The input document does not contain any Network Firewall firewall policy
#            resources
```
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a Network Firewall firewall policy resource
# And: 'StatelessFragmentDefaultActions' has not been provided in 'FirewallPolicy'
# Then: FAIL

# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a Network Firewall firewall policy resource
# And: 'StatelessFragmentDefaultActions' has been provided in 'FirewallPolicy' as an empty list
# Then: FAIL

# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a Network Firewall firewall policy resource
# And: 'StatelessFragmentDefaultActions' has been provided in 'FirewallPolicy' as a list that does not contain one of 'aws:drop' or 'aws:forward_to_sfe'
# Then: FAIL

# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a Network Firewall firewall policy resource
# And: 'StatelessFragmentDefaultActions' has been provided in 'FirewallPolicy' as a list that contains either 'aws:drop' or 'aws:forward_to_sfe'
# Then: PASS

# Constants

let NETFW_FIREWALL_POLICY_TYPE = "AWS::NetworkFirewall::FirewallPolicy"
let ALLOWED_STATELESS_FRAGMENT_ACTIONS_LIST = [ "aws:drop", "aws:forward_to_sfe" ]
let INPUT_DOCUMENT = this

# Assignments

let netfw_firewall_policies = Resources.*[ Type == %NETFW_FIREWALL_POLICY_TYPE ]

# Primary Rules

rule netfw_policy_default_action_fragment_packets_check when is_cfn_template(%INPUT_DOCUMENT)
  %netfw_firewall_policies not empty {
    check(%netfw_firewall_policies.Properties)
    <<
      [CT.NETWORK-FIREWALL.PR.3]: Require any AWS Network Firewall firewall policy to drop or forward fragmented packets by default when they do not match a stateless rule
      [FIX]: Within 'FirewallPolicy', include one of 'aws:drop' or 'aws:forward_to_sfe' in 'StatelessFragmentDefaultActions'.
    >>
  }

rule netfw_policy_default_action_fragment_packets_check when is_cfn_hook(%INPUT_DOCUMENT, %NETFW_FIREWALL_POLICY_TYPE) {
  check(%INPUT_DOCUMENT.%NETFW_FIREWALL_POLICY_TYPE.resourceProperties)
  <<
    [CT.NETWORK-FIREWALL.PR.3]: Require any AWS Network Firewall firewall policy to drop or forward fragmented packets by default when they do not match a stateless rule
[FIX]: Within 'FirewallPolicy', include one of 'aws:drop' or 'aws:forward_to_sfe' in 'StatelessFragmentDefaultActions'.

```plaintext
rule check(netfw_firewall_policy) {
    %netfw_firewall_policy {
        # Scenario 2
        FirewallPolicy exists
        FirewallPolicy is_struct

        FirewallPolicy {
            # Scenario 2 and 3
            StatelessFragmentDefaultActions exists
            StatelessFragmentDefaultActions is_list
            StatelessFragmentDefaultActions not empty

            # Scenario 4 and 5
            some StatelessFragmentDefaultActions[*] {
                this in %ALLOWED_STATELESS_FRAGMENT_ACTIONS_LIST
            }
        }
    }
}
```

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example - Use this template to verify a compliant resource creation.**

```plaintext
Resources:
FirewallPolicy:
    Type: AWS::NetworkFirewall::FirewallPolicy
    Properties:
        FirewallPolicyName:
            Fn::Sub: ${AWS::StackName}-example
FirewallPolicy:
    StatelessDefaultActions:
        - aws:forward_to_sfe
    StatelessFragmentDefaultActions:
        - aws:drop
```

**CT.NETWORK-FIREWALL.PR.3 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
FirewallPolicy:
  Type: AWS::NetworkFirewall::FirewallPolicy
  Properties:
    FirewallPolicyName:
      Fn::Sub: ${AWS::StackName}-example
    FirewallPolicy:
      StatelessDefaultActions:
        - aws:pass
      StatelessFragmentDefaultActions:
        - aws:pass
```

[CT.NETWORK-FIREWALL.PR.4] Require any AWS Network Firewall rule group to contain at least one rule

This control checks whether an AWS Network Firewall stateless rule group contains rules.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::NetworkFirewall::RuleGroup
- **AWS CloudFormation guard rule:** [CT.NETWORK-FIREWALL.PR.4 rule specification (p. 974)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.NETWORK-FIREWALL.PR.4 rule specification (p. 974)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.NETWORK-FIREWALL.PR.4 example templates (p. 976)]

Explanation

A rule group contains rules that define how your firewall processes traffic in your VPC. An empty, stateless rule group, when present in a firewall policy, might give the impression that the rule group will process traffic. However, when the stateless rule group is empty, it does not process traffic.

**Usage considerations**

- This control applies only to AWS Network Firewall stateless rule groups.

Remediation for rule failure


The examples that follow show how to implement this remediation.

**AWS Network Firewall Rule Group - Example**

AWS Network Firewall rule group configured with a stateless rule. The example is shown in JSON and in YAML.
JSON example

```json
{
  "NetworkFirewallRuleGroup": {
    "Type": "AWS::NetworkFirewall::RuleGroup",
    "Properties": {
      "RuleGroupName": {
        "Fn::Sub": "${AWS::StackName}-sample"
      },
      "Capacity": 100,
      "Description": "Sample rule group",
      "Type": "STATELESS",
      "RuleGroup": {
        "RulesSource": {
          "StatelessRulesAndCustomActions": {
            "StatelessRules": [
              {
                "RuleDefinition": {
                  "MatchAttributes": {
                    "Sources": [
                      {
                        "AddressDefinition": "0.0.0.0/0"
                      }
                    ],
                    "Destinations": [
                      {
                        "AddressDefinition": "10.0.0.0/8"
                      }
                    ],
                    "SourcePorts": [
                      {
                        "FromPort": 15000,
                        "ToPort": 30000
                      }
                    ],
                    "DestinationPorts": [
                      {
                        "FromPort": 443,
                        "ToPort": 443
                      }
                    ],
                    "Protocols": [6]
                  },
                  "Actions": [
                    "aws:forward_to_sfe"
                  ]
                },
                "Priority": 1
              }
            ]
          }
        }
      }
    }
  }
}
```

YAML example

```yaml
973
```
NetworkFirewallRuleGroup:
Type: AWS::NetworkFirewall::RuleGroup
Properties:
  RuleGroupName: !Sub '${AWS::StackName}-sample'
  Capacity: 100
  Description: Sample rule group
  Type: STATELESS
  RuleGroup:
    RulesSource:
      StatelessRulesAndCustomActions:
        StatelessRules:
          - RuleDefinition:
              MatchAttributes:
                Sources:
                  - AddressDefinition: '0.0.0.0/0'
                Destinations:
                  - AddressDefinition: 10.0.0.0/8
                SourcePorts:
                  - FromPort: 15000
                    ToPort: 30000
                DestinationPorts:
                  - FromPort: 443
                    ToPort: 443
                Protocols:
                  - 6
              Actions:
                - aws:forward_to_sfe
              Priority: 1

---

CT.NETWORK-FIREWALL.PR.4 rule specification

```plaintext
# ##########################
# Rule Specification     #
# ##########################

# Rule Identifier:
# netfw_stateless_rule_group_not_empty_check

# Description:
# This control checks whether an AWS Network Firewall stateless rule group contains rules.

# Reports on:
# AWS::NetworkFirewall::RuleGroup

# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain any Network Firewall rule group resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a Network Firewall rule group resource
```
And: 'Type' is not equal to 'STATELESS'
Then: SKIP

Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a Network Firewall rule group resource
And: 'Type' is 'STATELESS'
And: 'RuleGroup.RulesSource.StatelessRulesAndCustomActions' has not been provided
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a Network Firewall rule group resource
And: 'Type' is 'STATELESS'
And: 'RuleGroup.RulesSource.StatelessRulesAndCustomActions' has been provided
And: 'StatelessRules' has not been provided within 'StatelessRulesAndCustomActions' or has been provided with an empty list value
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains a Network Firewall rule group resource
And: 'Type' is 'STATELESS'
And: 'RuleGroup.RulesSource.StatelessRulesAndCustomActions' has been provided
And: 'StatelessRules' has been provided within 'StatelessRulesAndCustomActions' as a non-empty list value
Then: PASS

Constants
let NETFW_RULE_GROUP_TYPE = "AWS::NetworkFirewall::RuleGroup"
let INPUT_DOCUMENT = this

Assignments
let netfw_rule_group = Resources.*[ Type == %NETFW_RULE_GROUP_TYPE ]

Primary Rules

rule netfw_stateless_rule_group_not_empty_check when is_cfn_template(%INPUT_DOCUMENT) { %netfw_rule_group not empty {
  check(%netfw_rule_group.Properties)
  <<
  [CT.NETWORK-FIREWALL.PR.4]: Require any AWS Network Firewall rule group to contain at least one rule
  [FIX]: Provide one or more AWS Network Firewall stateless rules within the 'RuleGroup.RulesSource.StatelessRulesAndCustomActions.StatelessRules' property.
  >>
}
}

rule netfw_stateless_rule_group_not_empty_check when is_cfn_hook(%INPUT_DOCUMENT, %NETFW_RULE_GROUP_TYPE) { check(%INPUT_DOCUMENT.%NETFW_RULE_GROUP_TYPE.resourceProperties)
  <<
  [CT.NETWORK-FIREWALL.PR.4]: Require any AWS Network Firewall rule group to contain at least one rule
  [FIX]: Provide one or more AWS Network Firewall stateless rules within the 'RuleGroup.RulesSource.StatelessRulesAndCustomActions.StatelessRules' property.
  >>
}

# Parameterized Rules

rule check(netfw_rule_group) {
  %netfw_rule_group[
    # Scenario 2
    Type exists
    Type == "STATELESS"
  ] {
    # Scenario 3
    RuleGroup exists
    RuleGroup is_struct
    RuleGroup {
      RulesSource exists
      RulesSource is_struct
      RulesSource {
        StatelessRulesAndCustomActions exists
        StatelessRulesAndCustomActions is_struct
        StatelessRulesAndCustomActions {
          # Scenarios 4 and 5
          StatelessRules exists
          StatelessRules is_list
          StatelessRules not empty
        }
      }
    }
  }
}

# Utility Rules

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

## CT.NETWORK-FIREWALL.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:

NetworkFirewallRuleGroup:
  Type: AWS::NetworkFirewall::RuleGroup
  Properties:
    RuleGroupName:
      Fn::Sub: ${AWS::StackName}-example
    Capacity: 100
    Description: Example rule group
    Type: STATELESS
    RuleGroup:
      RulesSource:
        StatelessRulesAndCustomActions:
          StatelessRules:
- **RuleDefinition:**
  - **MatchAttributes:**
    - **Sources:**
      - **AddressDefinition:** 0.0.0.0/0
    - **Destinations:**
      - **AddressDefinition:** 10.0.0.0/8
    - **SourcePorts:**
      - **FromPort:** 15000
      - **ToPort:** 30000
    - **DestinationPorts:**
      - **FromPort:** 443
      - **ToPort:** 443
    - **Protocols:**
      - 6
    - **Actions:**
      - `aws:forward_to_sfe`
    - **Priority:** 1

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

**Resources:**

- **NetworkFirewallRuleGroup:**
  - **Type:** `AWS::NetworkFirewall::RuleGroup`
  - **Properties:**
    - **RuleGroupName:**
      - `Fn::Sub`:
        - `${AWS::StackName}-example`
    - **Capacity:** 100
    - **Description:** Example rule group
    - **Type:** `STATELESS`
    - **RuleGroup:**
      - **RulesSource:**
        - **StatelessRulesAndCustomActions:**
          - **StatelessRules:** []

---

**Amazon OpenSearch controls**

**Topics**

- [CT.OPENSEARCH.PR.1](#) Require an Elasticsearch domain to encrypt data at rest (p. 978)
- [CT.OPENSEARCH.PR.2](#) Require an Elasticsearch domain to be created in a user-specified Amazon VPC (p. 982)
- [CT.OPENSEARCH.PR.3](#) Require an Elasticsearch domain to encrypt data sent between nodes (p. 987)
- [CT.OPENSEARCH.PR.4](#) Require an Elasticsearch domain to send error logs to Amazon CloudWatch Logs (p. 992)
- [CT.OPENSEARCH.PR.5](#) Require an Elasticsearch domain to send audit logs to Amazon CloudWatch Logs (p. 999)
- [CT.OPENSEARCH.PR.6](#) Require an Elasticsearch domain to have zone awareness and at least three data nodes (p. 1007)
- [CT.OPENSEARCH.PR.7](#) Require an Elasticsearch domain to have at least three dedicated master nodes (p. 1012)
- [CT.OPENSEARCH.PR.8](#) Require an Elasticsearch Service domain to use TLSv1.2 (p. 1019)
- [CT.OPENSEARCH.PR.9](#) Require an Amazon OpenSearch Service domain to encrypt data at rest (p. 1024)
• [CT.OPENSEARCH.PR.10] Require an Amazon OpenSearch Service domain to be created in a user-specified Amazon VPC (p. 1028)
• [CT.OPENSEARCH.PR.11] Require an Amazon OpenSearch Service domain to encrypt data sent between nodes (p. 1033)
• [CT.OPENSEARCH.PR.12] Require an Amazon OpenSearch Service domain to send error logs to Amazon CloudWatch Logs (p. 1038)
• [CT.OPENSEARCH.PR.13] Require an Amazon OpenSearch Service domain to send audit logs to Amazon CloudWatch Logs (p. 1045)
• [CT.OPENSEARCH.PR.14] Require an Amazon OpenSearch Service domain to have zone awareness and at least three data nodes (p. 1053)
• [CT.OPENSEARCH.PR.15] Require an Amazon OpenSearch Service domain to use fine-grained access control (p. 1058)
• [CT.OPENSEARCH.PR.16] Require an Amazon OpenSearch Service domain to use TLSv1.2 (p. 1064)

[CT.OPENSEARCH.PR.1] Require an Elasticsearch domain to encrypt data at rest

This control checks whether Elasticsearch domains have encryption-at-rest enabled.

• Control objective: Encrypt data at rest
• Implementation: AWS CloudFormation guard rule
• Control behavior: Proactive
• Resource types: AWS::Elasticsearch::Domain
• AWS CloudFormation guard rule: [CT.OPENSEARCH.PR.1 rule specification (p. 979)]

Details and examples

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.OPENSEARCH.PR.1 rule specification (p. 979)]
• For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.OPENSEARCH.PR.1 example templates (p. 981)]

Explanation

For an added layer of security for your sensitive data in OpenSearch, you should configure your OpenSearch to be encrypted at rest. Elasticsearch domains offer encryption of data at rest. The feature uses AWS KMS to store and manage your encryption keys. To perform the encryption, it uses the Advanced Encryption Standard algorithm with 256-bit keys (AES-256).

Remediation for rule failure

Within EncryptionAtRestOptions, set Enabled to true.

The examples that follow show how to implement this remediation.

Elasticsearch Domain - Example

An Elasticsearch domain configured with encryption-at-rest enabled. The example is shown in JSON and YAM.

JSON example

```json
{
    "ElasticsearchDomain": {
        "Type": "AWS::Elasticsearch::Domain",
```
"Properties": {
  "ElasticsearchVersion": 7.1,
  "ElasticsearchClusterConfig": {
    "InstanceCount": "1",
    "InstanceType": "t3.small.elasticsearch"
  },
  "EBSOptions": {
    "EBSEnabled": true,
    "Iops": "3000",
    "VolumeSize": "10",
    "VolumeType": "gp3"
  },
  "AccessPolicies": {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Deny",
        "Principal": {
          "AWS": "*"
        },
        "Action": "es:*",
        "Resource": "*"
      }
    ],
    "EncryptionAtRestOptions": {
      "Enabled": true
    }
  }
}

YAML example

ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
Properties:
  ElasticsearchVersion: 7.1
  ElasticsearchClusterConfig:
    InstanceCount: '1'
    InstanceType: t3.small.elasticsearch
  EBSOptions:
    EBSEnabled: true
    Iops: '3000'
    VolumeSize: '10'
    VolumeType: gp3
  AccessPolicies:
    Version: '2012-10-17'
    Statement:
      - Effect: Deny
        Principal:
          AWS: '*'
        Action: es:*'
        Resource: '*'
  EncryptionAtRestOptions:
    Enabled: true

CT.OPENSEARCH.PR.1 rule specification
# Proactive controls

## Rule Specification

# Rule Identifier:
#   elasticsearch_encrypted_at_rest_check

# Description:
#   This control checks whether Elasticsearch domains have encryption-at-rest enabled.

# Reports on:
#   AWS::Elasticsearch::Domain

# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
#   None

# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any Elasticsearch domain resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'EncryptionAtRestOptions' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'EncryptionAtRestOptions' has been provided
#     And: 'Enabled' in 'EncryptionAtRestOptions' has not been provided or provided and set to a value other than bool(true)
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'EncryptionAtRestOptions' has been provided
#     And: 'Enabled' in 'EncryptionAtRestOptions' has been provided and set to bool(true)
#     Then: PASS

# Constants

let ELASTICSEARCH_DOMAIN_TYPE = "AWS::Elasticsearch::Domain"
let INPUT_DOCUMENT = this

# Assignments

let elasticsearch_domains = Resources.*[ Type == %ELASTICSEARCH_DOMAIN_TYPE ]

# Primary Rules

rule elasticsearch_encrypted_at_rest_check when is_cfn_template(%INPUT_DOCUMENT)
  %elasticsearch_domains not empty {
    check(%elasticsearch_domains.Properties)
    "[CT.OPENSEARCH.PR.1]: Require an Elasticsearch domain to encrypt data at rest"
[FIX]: Within 'EncryptionAtRestOptions', set 'Enabled' to 'true'.

rule elasticsearch_encrypted_at_rest_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICSEARCH_DOMAIN_TYPE) {
  check(%INPUT_DOCUMENT.%ELASTICSEARCH_DOMAIN_TYPE.resourceProperties)
  <<
    [CT.OPENSEARCH.PR.1]: Require an Elasticsearch domain to encrypt data at rest
  >>
}

# Parameterized Rules
#
rule check(elasticsearch_domain) {
  %elasticsearch_domain {
    # Scenario 2
    EncryptionAtRestOptions exists
    EncryptionAtRestOptions is_struct
      EncryptionAtRestOptions {
        # Scenarios 3 and 4
        Enabled exists
        Enabled == true
      }
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

**CT.OPENSEARCH.PR.1 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

Resources:
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
Properties:
  ElasticsearchVersion: 7.1
  ElasticsearchClusterConfig:
    InstanceCount: '1'
    InstanceType: t3.small.elasticsearch
  EBSOptions:
    EBSEnabled: true
    Iops: '3000'
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.elasticsearch
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'
    EncryptionAtRestOptions:
      Enabled: false

[CT.OPENSEARCH.PR.2] Require an Elasticsearch domain to be created in a user-specified Amazon VPC

This control checks whether Elasticsearch domains are configured with VPC option settings that specify a target Amazon VPC.

• Control objective: Limit network access
• Implementation: AWS CloudFormation guard rule
• Control behavior: Proactive
• Resource types: AWS::Elasticsearch::Domain
• AWS CloudFormation guard rule: CT.OPENSEARCH.PR.2 rule specification (p. 984)

Details and examples
• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.OPENSEARCH.PR.2 rule specification (p. 984)
For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: CT.OPENSEARCH.PR.2 example templates (p. 986)

**Explanation**

Elasticsearch domains deployed within a VPC can communicate with VPC resources over the private AWS network, without the need to traverse the public internet. This configuration increases the security posture by limiting access to the data in transit. VPCs provide a number of network controls that help create secure access to Elasticsearch domains, including network ACLs and security groups. Security Hub recommends that you migrate public Elasticsearch domains to VPCs to take advantage of these controls.

**Remediation for rule failure**

Within VPCOptions, set SubnetIds to a list with one or more Amazon EC2 subnet IDs.

The examples that follow show how to implement this remediation.

**Elasticsearch Domain - Example**

An Elasticsearch domain configured to deploy within an Amazon VPC by means of VPC option settings. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "ElasticsearchDomain": {
    "Type": "AWS::Elasticsearch::Domain",
    "Properties": {
      "ElasticsearchVersion": 7.1,
      "ElasticsearchClusterConfig": {
        "InstanceCount": "1",
        "InstanceType": "t3.small.elasticsearch"
      },
      "EBSOptions": {
        "EBSEnabled": true,
        "Iops": "3000",
        "VolumeSize": "10",
        "VolumeType": "gp3"
      },
      "AccessPolicies": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Deny",
            "Principal": {
              "AWS": "*"
            },
            "Action": "es:*",
            "Resource": "*"
          }
        ],
        "VPCOptions": {
          "SubnetIds": [
            {
              "Ref": "Subnet"
            }
          ]
        }
      }
    }
  }
}
```
YAML example

ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.elasticsearch
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'
    VPCOptions:
      SubnetIds:
        - !Ref 'Subnet'

CT.OPENSEARCH.PR.2 rule specification

# # Rule Specification #
## Rule Identifier:
#   elasticsearch_in_vpc_only_check
## Description:
#   This control checks whether Elasticsearch domains are configured with VPC option settings that specify a target Amazon VPC.
## Reports on:
#   AWS::Elasticsearch::Domain
## Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
## Rule Parameters:
#   None
## Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any Elasticsearch domain resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document

984
And: The input document contains an Elasticsearch domain resource
And: 'VPCOptions' has not been provided
Then: FAIL
Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elasticsearch domain resource
And: 'VPCOptions' has been provided
And: 'SubnetIds' in 'VPCOptions' has not been provided or has been provided as an empty list
Then: FAIL
Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elasticsearch domain resource
And: 'VPCOptions' has been provided
And: 'SubnetIds' in 'VPCOptions' has been provided as a list with one or more values
Then: PASS

Constants
let ELASTICSEARCH_DOMAIN_TYPE = "AWS::Elasticsearch::Domain"
let INPUT_DOCUMENT = this

Assignments
let elasticsearch_domains = Resources.*[ Type == %ELASTICSEARCH_DOMAIN_TYPE ]

Primary Rules
rule elasticsearch_in_vpc_only_check when is_cfn_template(%INPUT_DOCUMENT)
%elasticsearch_domains not empty {
  check(%elasticsearch_domains.Properties)
  %elasticsearch_domains not empty {
    [CT.OPENSEARCH.PR.2]: Require an Elasticsearch domain to be created in a user-specified Amazon VPC
    [FIX]: Within 'VPCOptions', set 'SubnetIds' to a list with one or more Amazon EC2 subnet IDs.
  } |
}
rule elasticsearch_in_vpc_only_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICSEARCH_DOMAIN_TYPE) {
  check(%INPUT_DOCUMENT.%ELASTICSEARCH_DOMAIN_TYPE.resourceProperties)
  %elasticsearch_domains not empty {
    [CT.OPENSEARCH.PR.2]: Require an Elasticsearch domain to be created in a user-specified Amazon VPC
    [FIX]: Within 'VPCOptions', set 'SubnetIds' to a list with one or more Amazon EC2 subnet IDs.
  } |
}

Parameterized Rules
rule check(elasticsearch_domain) {
  %elasticsearch_domain {
    # Scenario 2
    VPCOptions exists
    VPCOptions is_struct
    VPCOptions {
# Scenarios 3 and 4
SubnetIds exists
SubnetIds is_list
SubnetIds not empty

# Utility Rules

```perl
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
```

```perl
rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

CT.OPENSEARCH.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:

VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16

Subnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId:
    Ref: VPC
    CidrBlock: 10.0.0.0/16
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: ''

ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.elasticsearch
    EBSOptions:
      EBSEnabled: true
      Iops: '5000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
Properties:
  ElasticsearchVersion: 7.1
  ElasticsearchClusterConfig:
    InstanceCount: '1'
    InstanceType: t3.small.elasticsearch
  EBSOptions:
    EBSEnabled: true
    Iops: '3000'
    VolumeSize: '10'
    VolumeType: gp3
  AccessPolicies:
    Version: '2012-10-17'
    Statement:
      - Effect: Deny
        Principal:
          AWS: '*'
        Action: es:*
        Resource: '*'

[CT.OPENSEARCH.PR.3] Require an Elasticsearch domain to encrypt data sent between nodes

This control checks whether Elasticsearch domains have node-to-node encryption enabled.

- **Control objective:** Encrypt data in transit
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Elasticsearch::Domain
- **AWS CloudFormation guard rule:** [CT.OPENSEARCH.PR.3 rule specification](p. 989)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.OPENSEARCH.PR.3 rule specification](p. 989)
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.OPENSEARCH.PR.3 example templates](p. 991)

Explanation

HTTPS (TLS) can help prevent potential attackers from eavesdropping on or manipulating network traffic using person-in-the-middle, or similar, attacks. Only encrypted connections over HTTPS (TLS) should
be allowed. Enabling node-to-node encryption for Elasticsearch domains ensures that intra-cluster communications are encrypted in transit.

**Usage considerations**

- A performance penalty may be associated with this configuration. You should be aware of and test the performance trade-offs before enabling this option.

**Remediation for rule failure**

Within `NodeToNodeEncryptionOptions`, set `Enabled` to `true`.

The examples that follow show how to implement this remediation.

**Elasticsearch Domain - Example**

An Elasticsearch domain configured with node-to-node encryption enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "ElasticsearchDomain": {
      "Type": "AWS::Elasticsearch::Domain",
      "Properties": {
         "ElasticsearchVersion": 7.1,
         "ElasticsearchClusterConfig": {
            "InstanceCount": "1",
            "InstanceType": "t3.small.elasticsearch"
         },
         "EBSOptions": {
            "EBSEnabled": true,
            "Iops": "3000",
            "VolumeSize": "10",
            "VolumeType": "gp3"
         },
         "AccessPolicies": {
            "Version": "2012-10-17",
            "Statement": [
               {
                  "Effect": "Deny",
                  "Principal": {
                     "AWS": "*"
                  },
                  "Action": "es:*",
                  "Resource": "*"
               }
            ]
         },
         "NodeToNodeEncryptionOptions": {
            "Enabled": true
         }
      }
   }
}
```

**YAML example**

```yaml
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceCount: 1
      InstanceType: t3.small.elasticsearch
    EBSOptions:
      EBSEnabled: true
      Iops: 3000
      VolumeSize: 10
      VolumeType: gp3
    AccessPolicies:
      Version: 2012-10-17
      Statement:
      - Effect: Deny
        Principal:
          AWS: *
        Action: es:*
        Resource: *
    NodeToNodeEncryptionOptions:
      Enabled: true
```
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.elasticsearch
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'
    NodeToNodeEncryptionOptions:
      Enabled: true

CT.OPENSEARCH.PR.3 rule specification

# ###################################
##       Rule Specification        ##
#####################################
#
# Rule Identifier:
#   elasticsearch_node_to_node_encryption_check
#
# Description:
#   This control checks whether Elasticsearch domains have node-to-node encryption enabled.
#
# Reports on:
#   AWS::Elasticsearch::Domain
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any Elasticsearch domain resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'NodeToNodeEncryptionOptions' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'NodeToNodeEncryptionOptions' has been provided
And: 'Enabled' in 'NodeToNodeEncryptionOptions' has not been provided or has been provided and set to a value other than bool(true)
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elasticsearch domain resource
And: 'NodeToNodeEncryptionOptions' has been provided
And: 'Enabled' in 'NodeToNodeEncryptionOptions' has been provided and set to a value of bool(true)
Then: PASS

Constants

let ELASTICSEARCH_DOMAIN_TYPE = "AWS::Elasticsearch::Domain"
let INPUT_DOCUMENT = this

Assignments

let elasticsearch_domains = Resources.*[ Type == %ELASTICSEARCH_DOMAIN_TYPE ]

Primary Rules

rule elasticsearch_node_to_node_encryption_check when is_cfn_template(%INPUT_DOCUMENT)
%elasticsearch_domains not empty {
    check(%elasticsearch_domains.Properties)
    %elasticsearch_domains not empty {
        [CT.OPENSEARCH.PR.3]: Require an Elasticsearch domain to encrypt data sent between nodes
        [FIX]: Within 'NodeToNodeEncryptionOptions', set 'Enabled' to 'true'.
    }
}

rule elasticsearch_node_to_node_encryption_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICSEARCH_DOMAIN_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTICSEARCH_DOMAIN_TYPE.resourceProperties)
    %elasticsearch_domains not empty {
        [CT.OPENSEARCH.PR.3]: Require an Elasticsearch domain to encrypt data sent between nodes
        [FIX]: Within 'NodeToNodeEncryptionOptions', set 'Enabled' to 'true'.
    }
}

Parameterized Rules

rule check(elasticsearch_domain) {
    %elasticsearch_domain {
        Scenario 2
        NodeToNodeEncryptionOptions exists
        NodeToNodeEncryptionOptions is_struct
        NodeToNodeEncryptionOptions {
            Scenarios 3 and 4
            Enabled exists
            Enabled == true
        }
    }
}

Utility Rules

990
# rule is_cfn_template(doc) {
%doc {
    AWSTemplateFormatVersion exists or
    Resources exists
}
}

# rule is_cfn_hook(doc, RESOURCE_TYPE) {
%doc.%RESOURCE_TYPE.resourceProperties exists
}

## CT.OPENSEARCH.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```
Resources:
    ElasticsearchDomain:
        Type: AWS::Elasticsearch::Domain
        Properties:
            ElasticsearchVersion: 7.1
            ElasticsearchClusterConfig:
                InstanceCount: '1'
                InstanceType: t3.small.elasticsearch
            EBSOptions:
                EBSEnabled: true
                Iops: '3000'
                VolumeSize: '10'
                VolumeType: gp3
            AccessPolicies:
                Version: '2012-10-17'
                Statement:
                    - Effect: Deny
                    Principal:
                        AWS: '*'
                    Action: es:*
                    Resource: '*'
            NodeToNodeEncryptionOptions:
                Enabled: true
```

**FAIL Example** - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
    ElasticsearchDomain:
        Type: AWS::Elasticsearch::Domain
        Properties:
            ElasticsearchVersion: 7.1
            ElasticsearchClusterConfig:
                InstanceCount: '1'
                InstanceType: t3.small.elasticsearch
            EBSOptions:
                EBSEnabled: true
                Iops: '3000'
                VolumeSize: '10'
                VolumeType: gp3
```
Proactive controls

AccessPolicies:
  Version: '2012-10-17'
  Statement:
    - Effect: Deny
    Principal:
      AWS: '*'
    Action: es:*
    Resource: '*'
    NodeToNodeEncryptionOptions: {}
"ElasticsearchVersion": 7.1,
"ElasticsearchClusterConfig": {
  "InstanceCount": "1",
  "InstanceType": "t3.small.elasticsearch"
},
"EBSOptions": {
  "EBSEnabled": true,
  "Iops": "3000",
  "VolumeSize": "10",
  "VolumeType": "gp3"
},
"AccessPolicies": {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Resource": "*"
    }
  ],
  "LogPublishingOptions": {
    "ES_APPLICATION_LOGS": {
      "CloudWatchLogsLogGroupArn": {
        "Fn::GetAtt": [
          "LogGroup",
          "Arn"
        ],
        "Enabled": true
      }
    }
  }
}

YAML example

```yaml
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
Properties:
  ElasticsearchVersion: 7.1
  ElasticsearchClusterConfig:
    InstanceCount: '1'
    InstanceType: t3.small.elasticsearch
  EBSOptions:
    EBSEnabled: true
    Iops: '3000'
    VolumeSize: '10'
    VolumeType: gp3
  AccessPolicies:
    Version: '2012-10-17'
    Statement:
      - Effect: Deny
        Principal:
          AWS: '*'
        Action: es:*'
        Resource: '*'
  LogPublishingOptions:
    ES_APPLICATION_LOGS:
```

993
CT.OPENSEARCH.PR.4 rule specification

# #######################################################################
# Rule Specification       #
# #######################################################################
#
# Rule Identifier:
#   elasticsearch_application_logging_enabled_check
#
# Description:
#   This control checks whether Elasticsearch domains are configured to send error logs to
#   an Amazon CloudWatch Logs log group.
#
# Reports on:
#   AWS::Elasticsearch::Domain
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document does not contain any Elasticsearch domain resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'LogPublishingOptions' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'LogPublishingOptions' has been provided
#     And: 'ES_APPLICATION_LOGS' in 'LogPublishingOptions' has not been provided
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'LogPublishingOptions' has been specified
#     And: 'ES_APPLICATION_LOGS' in 'LogPublishingOptions' has been provided
#     And: 'Enabled' in 'ES_APPLICATION_LOGS' has not been provided or provided and set
#     to
#     a value other than bool(true)
#     And: 'CloudWatchLogsLogGroupArn' in 'ES_APPLICATION_LOGS' has not been provided or
#     provided
#     as an empty string or invalid local reference
#     Then: FAIL
#   Scenario: 5
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document contains an Elasticsearch domain resource

CloudWatchLogsLogGroupArn: !GetAtt 'LogGroup.Arn'
Enabled: true
And: 'LogPublishingOptions' has been specified
And: 'ES_APPLICATION_LOGS' in 'LogPublishingOptions' has been provided
And: 'Enabled' in 'ES_APPLICATION_LOGS' has been provided and set to bool(true)
And: 'CloudWatchLogsLogGroupArn' in 'ES_APPLICATION_LOGS' has not been provided or provided
as an empty string or invalid local reference
Then: FAIL
Scenarios:
1. Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elasticsearch domain resource
And: 'LogPublishingOptions' has been specified
And: 'ES_APPLICATION_LOGS' in 'LogPublishingOptions' has been provided
And: 'Enabled' in 'ES_APPLICATION_LOGS' has been provided and set to a value other than bool(true)
And: 'CloudWatchLogsLogGroupArn' in 'ES_APPLICATION_LOGS' has been provided as a non-empty string
or valid local reference
Then: FAIL
2. Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Elasticsearch domain resource
And: 'LogPublishingOptions' has been specified
And: 'ES_APPLICATION_LOGS' in 'LogPublishingOptions' has been provided
And: 'Enabled' in 'ES_APPLICATION_LOGS' has been provided and set to bool(true)
And: 'CloudWatchLogsLogGroupArn' in 'ES_APPLICATION_LOGS' has been provided as a non-empty string
or valid local reference
Then: PASS

Constants
let ELASTICSEARCH_DOMAIN_TYPE = "AWS::Elasticsearch::Domain"
let INPUT_DOCUMENT = this

Assignments
let elasticsearch_domains = Resources.*[ Type == %ELASTICSEARCH_DOMAIN_TYPE ]

Primary Rules
rule elasticsearch_application_logging_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%elasticsearch_domains not empty
{ check(%elasticsearch_domains.Properties)
  %elasticsearch_domains not empty
  <<
  [CT.OPENSEARCH.PR.4]: Require an Elasticsearch domain to send error logs to Amazon CloudWatch Logs
  [FIX]: Within 'LogPublishingOptions', provide an 'ES_APPLICATION_LOGS' configuration, set 'Enabled' to 'true', and set 'CloudWatchLogsLogGroupArn' to the ARN of a valid Amazon CloudWatch Logs log group.
  >>
}

rule elasticsearch_application_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT,
%ELASTICSEARCH_DOMAIN_TYPE) {
  check(%INPUT_DOCUMENT.%ELASTICSEARCH_DOMAIN_TYPE.resourceProperties)
  <<
  [CT.OPENSEARCH.PR.4]: Require an Elasticsearch domain to send error logs to Amazon CloudWatch Logs
}
[FIX]: Within 'LogPublishingOptions', provide an 'ES_APPLICATION_LOGS' configuration, set 'Enabled' to 'true', and set 'CloudWatchLogsLogGroupArn' to the ARN of a valid Amazon CloudWatch Logs log group.

```
rule check(elasticsearch_domain) {
  %elasticsearch_domain {
    LogPublishingOptions exists
    LogPublishingOptions is_struct
    LogPublishingOptions {
      ES_APPLICATION_LOGS exists
      ES_APPLICATION_LOGS is_struct
      ES_APPLICATION_LOGS {
        Enabled exists
        Enabled == true
        CloudWatchLogsLogGroupArn exists
        check_is_string_and_not_empty(CloudWatchLogsLogGroupArn) or
        check_local_references(%INPUT_DOCUMENT, CloudWatchLogsLogGroupArn, "AWS::Logs::LogGroup")
      } %elasticsearch_domain
    }
  }
}
```

# Parameterized Rules

 rule check_local_references(doc, reference_properties, referenced_resource_type) {
%reference_properties {
'Fn::GetAtt' {
  query_for_resource(%doc, this[0], %referenced_resource_type) <<Local Stack reference was invalid>>
} or Ref {
  query_for_resource(%doc, this, %referenced_resource_type) <<Local Stack reference was invalid>>
} %reference_properties
```
rule query_for_resource(doc, resource_key, referenced_resource_type) {
    let referenced_resource = %doc.Resources[ keys == %resource_key ]
    %referenced_resource not empty
    %referenced_resource {
        Type == %referenced_resource_type
    }
}

**CT.OPENSEARCH.PR.4 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example - Use this template to verify a compliant resource creation.**

<table>
<thead>
<tr>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ElasticsearchDomain:</td>
</tr>
<tr>
<td>Type: AWS::Elasticsearch::Domain</td>
</tr>
<tr>
<td>DependsOn: LogGroupPolicy</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>ElasticsearchVersion: 7.1</td>
</tr>
<tr>
<td>ElasticsearchClusterConfig:</td>
</tr>
<tr>
<td>InstanceCount: '1'</td>
</tr>
<tr>
<td>InstanceType: t3.small.elasticsearch</td>
</tr>
<tr>
<td>EBSOptions:</td>
</tr>
<tr>
<td>EBSEnabled: true</td>
</tr>
<tr>
<td>Iops: '3000'</td>
</tr>
<tr>
<td>VolumeSize: '10'</td>
</tr>
<tr>
<td>VolumeType: gp3</td>
</tr>
<tr>
<td>AccessPolicies:</td>
</tr>
<tr>
<td>Version: '2012-10-17'</td>
</tr>
<tr>
<td>Statement:</td>
</tr>
<tr>
<td>- Effect: Deny</td>
</tr>
<tr>
<td>Principal:</td>
</tr>
<tr>
<td>AWS: '*'</td>
</tr>
<tr>
<td>Action: es:*</td>
</tr>
<tr>
<td>Resource: '*'</td>
</tr>
<tr>
<td>LogPublishingOptions:</td>
</tr>
<tr>
<td>ES_APPLICATION_LOGS:</td>
</tr>
<tr>
<td>CloudWatchLogsLogGroupArn:</td>
</tr>
<tr>
<td>Fn::GetAtt:</td>
</tr>
<tr>
<td>- LogGroup</td>
</tr>
<tr>
<td>- Arn</td>
</tr>
<tr>
<td>Enabled: true</td>
</tr>
<tr>
<td>LogGroup:</td>
</tr>
<tr>
<td>Type: AWS::Logs::LogGroup</td>
</tr>
<tr>
<td>LogGroupPolicy:</td>
</tr>
<tr>
<td>Type: AWS::Logs::ResourcePolicy</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>PolicyName:</td>
</tr>
<tr>
<td>Fn::Sub: ${AWS::StackName}-AllowES</td>
</tr>
<tr>
<td>PolicyDocument:</td>
</tr>
</tbody>
</table>
"${LogGroupArn}","Condition": {"StringEquals": {"aws:SourceAccount": 
"${AWS::AccountId}"}}}}', LogGroupArn: |
|   - LogGroupArn:     |
|       Fn::GetAtt: [ LogGroup, Arn ]
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.elasticsearch
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.elasticsearch
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'
    LogPublishingOptions:
      ES_APPLICATION_LOGS:
        Enabled: false
[CT.OPENSEARCH.PR.5] Require an Elasticsearch domain to send audit logs to Amazon CloudWatch Logs

This control checks whether Elasticsearch domains are configured to send audit logs to an Amazon CloudWatch Logs log group.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Elasticsearch::Domain
- **AWS CloudFormation guard rule:** [CT.OPENSEARCH.PR.5 rule specification](p. 1001)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.OPENSEARCH.PR.5 rule specification](p. 1001)
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.OPENSEARCH.PR.5 example templates](p. 1004)

**Explanation**

Audit logs are highly customizable. They allow you to track user activity on your Elasticsearch clusters, including authentication successes and failures, requests to OpenSearch, index changes, and incoming search queries.

**Usage considerations**

- This control requires that Elasticsearch domains must have advanced security options configured.
- To enable advanced security options on an Elasticsearch domain through the AdvancedSecurityOptions property, you must enable encryption of data at rest (by means of EncryptionAtRestOptions), node-to-node encryption (by means of NodeToNodeEncryptionOptions), and enforce HTTPS connections (by means of DomainEndpointOptions).

**Remediation for rule failure**

Within LogPublishingOptions, provide an AUDIT_LOGS configuration, set Enabled to true, and set CloudWatchLogsLogGroupArn to the ARN of a valid Amazon CloudWatch Logs log group.

The examples that follow show how to implement this remediation.

**Elasticsearch Domain - Example**

An Elasticsearch domain configured to send audit logs to Amazon CloudWatch Logs. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ElasticsearchDomain": {
        "Type": "AWS::Elasticsearch::Domain",
        "Properties": {
            "ElasticsearchVersion": 7.1,
            "ElasticsearchClusterConfig": {
```
"InstanceCount": "1",
"InstanceType": "t3.small.elasticsearch"
},
"EBSOptions": {
  "EBSEnabled": true,
  "Iops": "3000",
  "VolumeSize": "10",
  "VolumeType": "gp3"
},
"AccessPolicies": {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Resource": "*"
    }
  ]
},
"NodeToNodeEncryptionOptions": {
  "Enabled": true
},
"EncryptionAtRestOptions": {
  "Enabled": true
},
"DomainEndpointOptions": {
  "EnforceHTTPS": true
},
"AdvancedSecurityOptions": {
  "Enabled": true,
  "InternalUserDatabaseEnabled": false,
  "MasterUserOptions": {
    "MasterUserARN": {
      "Fn::GetAtt": [
        "IAMRole",
        "Arn"
      ]
    }
  }
},
"LogPublishingOptions": {
  "AUDIT_LOGS": {
    "CloudWatchLogsLogGroupArn": {
      "Fn::GetAtt": [
        "LogGroup",
        "Arn"
      ]
    },
    "Enabled": true
  }
}

YAML example

ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
ElasticsearchVersion: 7.1
ElasticsearchClusterConfig:
  InstanceCount: '1'
  InstanceType: t3.small.elasticsearch
EBSOptions:
  EBSEnabled: true
  Iops: '3000'
  VolumeSize: '10'
  VolumeType: gp3
AccessPolicies:
  Version: '2012-10-17'
  Statement:
    - Effect: Deny
      Principal:
        AWS: '*'
      Action: es:*
      Resource: '*'
NodeToNodeEncryptionOptions:
  Enabled: true
EncryptionAtRestOptions:
  Enabled: true
DomainEndpointOptions:
  EnforceHTTPS: true
AdvancedSecurityOptions:
  Enabled: true
  InternalUserDatabaseEnabled: false
  MasterUserOptions:
    MasterUserARN: !GetAtt 'IAMRole.Arn'
LogPublishingOptions:
  AUDIT_LOGS:
    CloudWatchLogsLogGroupArn: !GetAtt 'LogGroup.Arn'
    Enabled: true

CT.OPENSEARCH.PR.5 rule specification

# #####################################################################
## Rule Specification
# #####################################################################

# Rule Identifier:
#  elasticsearch_audit_logging_enabled_check
#
# Description:
#  This control checks whether Elasticsearch domains are configured to send audit logs to
#  an Amazon CloudWatch Logs log group.
#
# Reports on:
#  AWS::Elasticsearch::Domain
#
# Evaluates:
#  AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#  None
#
# Scenarios:
#  Scenario: 1
#  Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#  document
#  And: The input document does not contain any Elasticsearch domain resources
#  Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Elasticsearch domain resource
# And: 'LogPublishingOptions' has not been provided
# Then: FAIL

# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Elasticsearch domain resource
# And: 'LogPublishingOptions' has been provided
# And: 'AUDIT_LOGS' in 'LogPublishingOptions' has not been provided
# Then: FAIL

# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Elasticsearch domain resource
# And: 'LogPublishingOptions' has been specified
# And: 'AUDIT_LOGS' in 'LogPublishingOptions' has been provided
# And: 'Enabled' in 'AUDIT_LOGS' has not been provided or provided and set to
# a value other than bool(true)
# And: 'CloudWatchLogsLogGroupArn' in 'AUDIT_LOGS' has not been provided or provided
# as an empty string or invalid local reference
# Then: FAIL

# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Elasticsearch domain resource
# And: 'LogPublishingOptions' has been specified
# And: 'AUDIT_LOGS' in 'LogPublishingOptions' has been provided
# And: 'Enabled' in 'AUDIT_LOGS' has been provided and set to bool(true)
# And: 'CloudWatchLogsLogGroupArn' in 'AUDIT_LOGS' has not been provided or provided
# as an empty string or invalid local reference
# Then: FAIL

# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Elasticsearch domain resource
# And: 'LogPublishingOptions' has been specified
# And: 'AUDIT_LOGS' in 'LogPublishingOptions' has been provided
# And: 'Enabled' in 'AUDIT_LOGS' has not been provided or provided and set to
# a value other than bool(true)
# And: 'CloudWatchLogsLogGroupArn' in 'AUDIT_LOGS' has been provided as a non-empty
string
# or valid local reference
# Then: FAIL

# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Elasticsearch domain resource
# And: 'LogPublishingOptions' has been specified
# And: 'AUDIT_LOGS' in 'LogPublishingOptions' has been provided
# And: 'Enabled' in 'AUDIT_LOGS' has been provided and set to bool(true)
# And: 'CloudWatchLogsLogGroupArn' in 'AUDIT_LOGS' has been provided as a non-empty
string
# or valid local reference
# Then: PASS

# Constants
let ELASTICSEARCH_DOMAIN_TYPE = "AWS::Elasticsearch::Domain"
let INPUT_DOCUMENT = this
let elasticsearch_domains = Resources.*[ Type == %ELASTICSEARCH_DOMAIN_TYPE ]
#
# Primary Rules
#
rule elasticsearch_audit_logging_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
elasticsearch_domains not empty {
    check(elasticsearch_domains.Properties)
    <<
    [CT.OPENSEARCH.PR.5]: Require an Elasticsearch domain to send audit logs to Amazon CloudWatch Logs
    [FIX]: Within 'LogPublishingOptions', provide an 'AUDIT_LOGS' configuration, set 'Enabled' to 'true', and set 'CloudWatchLogsLogGroupArn' to the ARN of a valid Amazon CloudWatch Logs log group.
>>
}
rule elasticsearch_audit_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICSEARCH_DOMAIN_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTICSEARCH_DOMAIN_TYPE.resourceProperties)
    <<
    [CT.OPENSEARCH.PR.5]: Require an Elasticsearch domain to send audit logs to Amazon CloudWatch Logs
    [FIX]: Within 'LogPublishingOptions', provide an 'AUDIT_LOGS' configuration, set 'Enabled' to 'true', and set 'CloudWatchLogsLogGroupArn' to the ARN of a valid Amazon CloudWatch Logs log group.
>>
}
#
# Parameterized Rules
#
rule check(elasticsearch_domain) {
    elasticsearch_domain {
        # Scenario 2
        LogPublishingOptions exists
        LogPublishingOptions is_struct
        LogPublishingOptions {
            # Scenario 3
            AUDIT_LOGS exists
            AUDIT_LOGS is_struct
            AUDIT_LOGS {
                # Scenarios 4, 5, 6 and 7
                Enabled exists
                Enabled == true
                CloudWatchLogsLogGroupArn exists
                check_is_string_and_not_empty(CloudWatchLogsLogGroupArn) or
                check_local_references(%INPUT_DOCUMENT, CloudWatchLogsLogGroupArn, "AWS::Logs::LogGroup")
            }
        }
    }
}
#
# Utility Rules
#
rule is_cfn_template(doc) {
    doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
AWS Control Tower User Guide
Proactive controls

CT.OPENSEARCH.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
IAMRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            AWS:
              Ref: AWS::AccountId
          Action: sts:AssumeRole
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  DependsOn: LogGroupPolicy
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.elasticsearch
    EBSOptions:
      EBSEnabled: true
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
IAMRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            AWS: ${AWS::AccountId}
          Action: sts:AssumeRole
ElasticsearchDomain:
Type: AWS::Elasticsearch::Domain
Properties:
  ElasticsearchVersion: 7.1
  ElasticsearchClusterConfig:
    InstanceCount: '1'
    InstanceType: t3.small.elasticsearch
EBSOPTIONS:
  EBS{Enabled, true
  Iops: '3000'
  VolumeSize: '10'
  VolumeType: gp3
AccessPolicies:
  Version: '2012-10-17'
  Statement:
    - Effect: Deny
      Principal:
        AWS: '*'
      Action: es:*
      Resource: '*'
NodeToNodeEncryptionOptions:
  Enabled: true
EncryptionAtRestOptions:
  Enabled: true
DomainEndpointOptions:
  EnforceHTTPS: true
AdvancedSecurityOptions:
  Enabled: true
  InternalUserDatabaseEnabled: false
MasterUserOptions:
  MasterUserARN:
    Fn::GetAtt:
    - IAMRole
    - Arn

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
IAMRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            AWS:
              Ref: AWS::AccountId
          Action: sts:AssumeRole
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.elasticsearch
EBSOPTIONS:
  EBS{Enabled, true
  Iops: '3000'
  VolumeSize: '10'
  VolumeType: gp3
AccessPolicies:
  Version: '2012-10-17'
[CT.OPENSEARCH.PR.6] Require an Elasticsearch domain to have zone awareness and at least three data nodes

This control checks whether ElasticSearch domains are configured with at least three data nodes and zone awareness enabled.

- **Control objective:** Improve availability
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Elasticsearch::Domain
- **AWS CloudFormation guard rule:** [CT.OPENSEARCH.PR.6 rule specification (p. 1009)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.OPENSEARCH.PR.6 rule specification (p. 1009)]
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.OPENSEARCH.PR.6 example templates (p. 1011)]

Explanation

An Elasticsearch domain requires at least three data nodes for high availability and fault-tolerance. Deploying an Elasticsearch domain with at least three data nodes ensures that cluster operations can continue if a node fails.

Remediation for rule failure

Within ElasticsearchClusterConfig, set ZoneAwarenessEnabled to true, and set InstanceCount to an integer value greater than or equal to three.

The examples that follow show how to implement this remediation.
Elasticsearch Domain - Example

An Elasticsearch domain configured with three data nodes and zone awareness enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "ElasticsearchDomain": {
      "Type": "AWS::Elasticsearch::Domain",
      "Properties": {
         "ElasticsearchVersion": 7.1,
         "EBSOptions": {
            "EBSEnabled": true,
            "Iops": "3000",
            "VolumeSize": "10",
            "VolumeType": "gp3"
         },
         "AccessPolicies": {
            "Version": "2012-10-17",
            "Statement": [
               {
                  "Effect": "Deny",
                  "Principal": {"AWS": "*"},
                  "Action": "es:*",
                  "Resource": "*"
               }
            ]
         }
      },
      "ElasticsearchClusterConfig": {
         "InstanceType": "t3.small.elasticsearch",
         "InstanceCount": 3,
         "ZoneAwarenessEnabled": true,
         "ZoneAwarenessConfig": {
            "AvailabilityZoneCount": 3
         }
      }
   }
}
```

**YAML example**

```yaml
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*```

1008
Resource: '*'
ElasticsearchClusterConfig:
  InstanceType: t3.small.elasticsearch
  InstanceCount: 3
  ZoneAwarenessEnabled: true
  ZoneAwarenessConfig:
    AvailabilityZoneCount: 3

CT.OPENSEARCH.PR.6 rule specification

```
# ####################################################################
#                    Rule Specification                               
# ####################################################################
#
# Rule Identifier:
#   elasticsearch_data_node_fault_tolerance_check
#
# Description:
#   This control checks whether Elasticsearch domains are configured with at least three
#   data nodes and zone awareness enabled.
#
# Reports on:
#   AWS::Elasticsearch::Domain
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any Elasticsearch domain resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'ElasticsearchClusterConfig' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'ElasticsearchClusterConfig' has been provided
#     And: 'ZoneAwarenessEnabled' in 'ElasticsearchClusterConfig' has not been provided
#     or provided and set to a value other than bool(true)
#     And: 'InstanceCount' in 'ElasticsearchClusterConfig' has not been provided or
#     provided and set to an integer value less than three (< 3)
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'ElasticsearchClusterConfig' has been provided
#     And: 'ZoneAwarenessEnabled' in 'ElasticsearchClusterConfig' has been provided
#     and set to bool(true)
#     And: 'InstanceCount' in 'ElasticsearchClusterConfig' has not been provided or
#     provided and set to an integer value less than three (< 3)
```
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Elasticsearch domain resource
# And: 'ElasticsearchClusterConfig' has been provided
# And: 'ZoneAwarenessEnabled' in 'ElasticsearchClusterConfig' has not been provided or provided and set to a value other than bool(true)
# And: 'InstanceCount' in 'ElasticsearchClusterConfig' has been provided and set to an integer value greater than or equal to three (>= 3)
# Then: FAIL

# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an Elasticsearch domain resource
# And: 'ElasticsearchClusterConfig' has been provided
# And: 'ZoneAwarenessEnabled' in 'ElasticsearchClusterConfig' has been provided and set to bool(true)
# And: 'InstanceCount' in 'ElasticsearchClusterConfig' has been provided and set to an integer value greater than or equal to three (>= 3)
# Then: PASS

# Constants
#
let ELASTICSEARCH_DOMAIN_TYPE = "AWS::Elasticsearch::Domain"
let INPUT_DOCUMENT = this
#
# Assignments
#
let elasticsearch_domains = Resources.*[ Type == %ELASTICSEARCH_DOMAIN_TYPE ]
#
# Primary Rules
#
rule elasticsearch_data_node_fault_tolerance_check when is_cfn_template(%INPUT_DOCUMENT) {
    %elasticsearch_domains not empty {
        check(%elasticsearch_domains.Properties)
        <<<
        [CT.OPENSEARCH.PR.6]: Require an Elasticsearch domain to have zone awareness and at least three data nodes
        [FIX]: Within 'ElasticsearchClusterConfig', set 'ZoneAwarenessEnabled' to 'true', and set 'InstanceCount' to an integer value greater than or equal to three.
        >>>
    }
}

rule elasticsearch_data_node_fault_tolerance_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICSEARCH_DOMAIN_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTICSEARCH_DOMAIN_TYPE.resourceProperties)
    <<<
    [CT.OPENSEARCH.PR.6]: Require an Elasticsearch domain to have zone awareness and at least three data nodes
    [FIX]: Within 'ElasticsearchClusterConfig', set 'ZoneAwarenessEnabled' to 'true', and set 'InstanceCount' to an integer value greater than or equal to three.
    >>>
}

# Parameterized Rules
#
rule check(elasticsearch_domain) {
    %elasticsearch_domain {
        # Scenario 2
        ElasticsearchClusterConfig exists
        ElasticsearchClusterConfig is_struct
ElasticsearchClusterConfig {
  # Scenario 3, 4, 5 and 6
  ZoneAwarenessEnabled exists
  ZoneAwarenessEnabled == true
    InstanceCount exists
    InstanceCount >= 3
  }
}

# Utility Rules
#
# rule is_cfn_template(doc) {
#   %doc {
#       AWSTemplateFormatVersion exists or
#       Resources exists
#   }
# }
#
# rule is_cfn_hook(doc, RESOURCE_TYPE) {
#   %doc.%RESOURCE_TYPE.resourceProperties exists
# }

CT.OPENSEARCH.PR.6 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
Properties:
  ElasticsearchVersion: 7.1
EBSE Options:
  EBSEEnabled: true
Iops: '3000'
VolumeSize: '10'
VolumeType: gp3
AccessPolicies:
  Version: '2012-10-17'
  Statement:
    - Effect: Deny
      Principal:
        AWS: '*'
      Action: es:*
      Resource: '*'
ElasticsearchClusterConfig:
  InstanceType: t3.small.elasticsearch
  InstanceCount: 3
  ZoneAwarenessEnabled: true
  ZoneAwarenessConfig:
    AvailabilityZoneCount: 3

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.
[CT.OPENSEARCH.PR.7] Require an Elasticsearch domain to have at least three dedicated master nodes

This control checks whether Elasticsearch domains are configured with at least three dedicated master nodes.

- **Control objective:** Improve availability  
- **Implementation:** AWS CloudFormation guard rule  
- **Control behavior:** Proactive  
- **Resource types:** AWS::Elasticsearch::Domain  
- **AWS CloudFormation guard rule:** [CT.OPENSEARCH.PR.7 rule specification (p. 1015)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.OPENSEARCH.PR.7 rule specification (p. 1015)]  
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.OPENSEARCH.PR.7 example templates (p. 1017)]

**Explanation**

An Elasticsearch domain requires at least three dedicated master nodes for high availability and fault-tolerance. Dedicated master node resources can be strained during data node blue/green deployments, because additional nodes must be managed. Deploying an Elasticsearch domain with at least three dedicated master nodes ensures that sufficient master node resource capacity exists, and that cluster operations can continue if a node fails.

**Remediation for rule failure**

Within ElasticsearchClusterConfig, set DedicatedMasterEnabled to true, and set DedicatedMasterCount to an integer value greater than or equal to three, or omit the DedicatedMasterCount property to adopt the default value of three.
The examples that follow show how to implement this remediation.

**Elasticsearch Domain - Example One**

An Elasticsearch domain configured with three dedicated master nodes by means of the `DedicatedMasterCount` property. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "ElasticsearchDomain": {
      "Type": "AWS::Elasticsearch::Domain",
      "Properties": {
         "ElasticsearchVersion": 7.1,
         "EBSOptions": {
            "EBSEnabled": true,
            "VolumeSize": "10",
            "VolumeType": "gp3"
         },
         "AccessPolicies": {
            "Version": "2012-10-17",
            "Statement": [
               {
                  "Effect": "Deny",
                  "Principal": {
                     "AWS": "*
                  },
                  "Action": "es:*",
                  "Resource": "*
               }
            ]
         },
         "ElasticsearchClusterConfig": {
            "InstanceType": "t3.small.elasticsearch",
            "DedicatedMasterEnabled": true,
            "DedicatedMasterCount": 3
         }
      }
   }
}
```

**YAML example**

```yaml
ElasticsearchDomain:
    Type: AWS::Elasticsearch::Domain
    Properties:
        ElasticsearchVersion: 7.1
        EBSOptions:
            EBSEnabled: true
            Iops: '3000'
            VolumeSize: '10'
            VolumeType: gp3
        AccessPolicies:
            Version: '2012-10-17'
            Statement:
            - Effect: Deny
              Principal:
                AWS: '*'
              Action: es:*
The examples that follow show how to implement this remediation.

**Elasticsearch Domain - Example Two**

An Elasticsearch domain configured with three dedicated master nodes by means of the AWS CloudFormation defaults. The example is shown in JSON and in YAML.

**JSON example**

```
{
  "ElasticsearchDomain": {
    "Type": "AWS::Elasticsearch::Domain",
    "Properties": {
      "ElasticsearchVersion": 7.1,
      "EBSOptions": { 
        "EBSEnabled": true,
        "Iops": "3000",
        "VolumeSize": "10",
        "VolumeType": "gp3"
      },
      "AccessPolicies": { 
        "Version": "2012-10-17",
        "Statement": [ 
          { 
            "Effect": "Deny",
            "Principal": { 
              "AWS": "*"
            },
            "Action": "es:*",
            "Resource": "*"
          }
        ]
      },
      "ElasticsearchClusterConfig": { 
        "InstanceType": "t3.small.elasticsearch",
        "DedicatedMasterEnabled": true
      }
    }
  }
}
```

**YAML example**

```
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
```
AccessPolicies:
  Version: '2012-10-17'
  Statement:
    - Effect: Deny
    Principal:
      AWS: '*'
    Action: es:*
    Resource: '*'
ElasticsearchClusterConfig:
  InstanceType: t3.small.elasticsearch
  DedicatedMasterEnabled: true

CT.OPENSEARCH.PR.7 rule specification

# ####################################################################
##       Rule Specification        ##
# ####################################################################

# Rule Identifier:
#   elasticsearch_primary_node_fault_tolerance_check
#
# Description:
#   This control checks whether Elasticsearch domains are configured with at least three
dedicated master nodes.
#
# Reports on:
#   AWS::Elasticsearch::Domain
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any Elasticsearch domain resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document contains an Elasticsearch domain resource
#     And: 'ElasticsearchClusterConfig' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document contains an Elasticsearch domain resource
#     And: 'ElasticsearchClusterConfig' has been provided
#     And: 'DedicatedMasterEnabled' in 'ElasticsearchClusterConfig' has not been provided
#     Or provided and set to a value other than bool(true)
#     Or 'DedicatedMasterCount' in 'ElasticsearchClusterConfig' has not been provided
#     Or provided and set to an integer value less than three (< 3)
#     Then: FAIL
#   Scenario: 4
#     Given: The input document contains an Elasticsearch domain resource

1015
# And: 'ElasticsearchClusterConfig' has been provided
# And: 'DedicatedMasterEnabled' in 'ElasticsearchClusterConfig' has not been provided
# or provided and set to a value other than bool(true)
# And: 'DedicatedMasterCount' in 'ElasticsearchClusterConfig' has been provided and
# set to an integer value greater than or equal to three (>= 3)
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Elasticsearch domain resource
# And: 'ElasticsearchClusterConfig' has been provided
# And: 'DedicatedMasterEnabled' in 'ElasticsearchClusterConfig' has been provided
# and set to bool(true)
# And: 'DedicatedMasterCount' in 'ElasticsearchClusterConfig' has been provided and
# set to an integer value less than three (< 3)
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Elasticsearch domain resource
# And: 'ElasticsearchClusterConfig' has been provided
# And: 'DedicatedMasterEnabled' in 'ElasticsearchClusterConfig' has been provided
# and set to bool(true)
# And: 'DedicatedMasterCount' in 'ElasticsearchClusterConfig' has not been provided
# Then: PASS
# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an Elasticsearch domain resource
# And: 'ElasticsearchClusterConfig' has been provided
# And: 'DedicatedMasterEnabled' in 'ElasticsearchClusterConfig' has been provided
# and set to bool(true)
# And: 'DedicatedMasterCount' in 'ElasticsearchClusterConfig' has been provided and
# set to an integer value greater than or equal to three (>= 3)
# Then: PASS

# Constants
#
let ELASTICSEARCH_DOMAIN_TYPE = "AWS::Elasticsearch::Domain"
let INPUT_DOCUMENT = this

#
# Assignments
#
let elasticsearch_domains = Resources.*[ Type == %ELASTICSEARCH_DOMAIN_TYPE ]

#
# Primary Rules
#
rule elasticsearch_primary_node_fault_tolerance_check when is_cfn_template(%INPUT_DOCUMENT)
%elasticsearch_domains not empty
{
  check(%elasticsearch_domains.Properties)
  <<
  [CT.OPENSEARCH.PR.7]: Require an Elasticsearch domain to have at least three
dedicated master nodes
  [FIX]: Within 'ElasticsearchClusterConfig', set 'DedicatedMasterEnabled' to 'true',
and set 'DedicatedMasterCount' to an integer value greater than or equal to three,
or omit the 'DedicatedMasterCount' property to adopt the default value of three.
  >>
}
rule elasticsearch_primary_node_fault_tolerance_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICSEARCH_DOMAIN_TYPE) {
    check(%INPUT_DOCUMENT.%ELASTICSEARCH_DOMAIN_TYPE.resourceProperties)
    <<
    [CT.OPENSEARCH.PR.7]: Require an Elasticsearch domain to have at least three dedicated master nodes
    [FIX]: Within 'ElasticsearchClusterConfig', set 'DedicatedMasterEnabled' to 'true', and set 'DedicatedMasterCount' to an integer value greater than or equal to three, or omit the 'DedicatedMasterCount' property to adopt the default value of three.
    }
}
# Parameterized Rules
#
# Scenario 2
ElasticsearchClusterConfig exists
ElasticsearchClusterConfig is_struct
ElasticsearchClusterConfig {
    # Scenario 3, 4, 5 and 6
    DedicatedMasterEnabled exists
    DedicatedMasterEnabled == true
    DedicatedMasterCount not exists or DedicatedMasterCount >= 3
}
}
#
# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or Resources exists
    }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.OPENSEARCH.PR.7 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
ElasticsearchDomain:
    Type: AWS::Elasticsearch::Domain
    Properties:
        ElasticsearchVersion: 7.1
        ElasticsearchClusterConfig:
            InstanceType: t3.small.elasticsearch
            DedicatedMasterEnabled: true
        EBSOptions:
EBSEnabled: true
Iops: '3000'
VolumeSize: '10'
VolumeType: gp3
AccessPolicies:
  Version: '2012-10-17'
  Statement:
    - Effect: Deny
    Principal:
      AWS: '*'
    Action: es:*
    Resource: '*'

PASS Example - Use this template to verify a compliant resource creation.

Resources:
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceType: t3.small.elasticsearch
      DedicatedMasterEnabled: true
      DedicatedMasterCount: 3
    EBSEnabled: true
    Iops: '3000'
    VolumeSize: '10'
    VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
        Principal:
          AWS: '*'
        Action: es:*
        Resource: '*'

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceType: t3.small.elasticsearch
      DedicatedMasterEnabled: false
    EBSEnabled: true
    Iops: '3000'
    VolumeSize: '10'
    VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
[CT.OPENSEARCH.PR.8] Require an Elasticsearch Service domain to use TLSv1.2

This control checks whether Elasticsearch Service domains are configured to require HTTPS with a minimum TLS version of TLSv1.2.

- **Control objective**: Encrypt data in transit
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::Elasticsearch::Domain
- **AWS CloudFormation guard rule**: CT.OPENSEARCH.PR.8 rule specification (p. 1020)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.OPENSEARCH.PR.8 rule specification (p. 1020)
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: CT.OPENSEARCH.PR.8 example templates (p. 1023)

Explanation

HTTPS (TLS) can help prevent potential attackers from using person-in-the-middle, or similar attacks, to eavesdrop on or manipulate network traffic. Only encrypted connections over HTTPS (TLS) should be allowed. Encrypting data in transit can affect performance. You should test your application with this feature to understand the performance profile and the effects of TLS. TLS 1.2 provides several security enhancements over previous versions of TLS.

Remediation for rule failure


The examples that follow show how to implement this remediation.

**Elasticsearch Domain - Example**

An Elasticsearch domain configured to require that all traffic to the domain arrives over HTTPS with a minimum TLS version of TLSv1.2. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "ElasticsearchDomain": {
        "Type": "AWS::Elasticsearch::Domain",
        "Properties": {
            "ElasticsearchVersion": 7.1,
            "ElasticsearchClusterConfig": {
                "InstanceCount": "1",
                "InstanceType": "t3.small.elasticsearch"
            }
        }
    }
}
```
"EBSOptions": { 
  "EBSEnabled": true,
  "Iops": "3000",
  "VolumeSize": "10",
  "VolumeType": "gp3"
},
"AccessPolicies": { 
  "Version": "2012-10-17",
  "Statement": [ 
    { 
      "Effect": "Deny",
      "Principal": { 
        "AWS": "*"
      },
      "Action": "es:*",
      "Resource": "*"
    }
  ],
  "DomainEndpointOptions": { 
    "EnforceHTTPS": true,
    "TLSSecurityPolicy": "Policy-Min-TLS-1-2-2019-07"
  }
}

YAML example

ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.elasticsearch
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal: 
            AWS: '*
          Action: es:*'
          Resource: '*'
    DomainEndpointOptions:
      EnforceHTTPS: true

CT.OPENSEARCH.PR.8 rule specification

# #######################################################################
##       Rule Specification        
########################################################################

1020
# Rule Identifier:
#   elasticsearch_https_required_check

# Description:
#   This control checks whether Elasticsearch domains are configured to require HTTPS with a minimum TLS version of TLSv1.2.

# Reports on:
#   AWS::Elasticsearch::Domain

# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
#   None

# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any Elasticsearch domain resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'DomainEndpointOptions' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'DomainEndpointOptions' has been provided
#     And: 'EnforceHTTPS' in 'DomainEndpointOptions' has not been provided or has been provided and set to a value other than bool(true)
#     And: 'TLSSecurityPolicy' in 'DomainEndpointOptions' has not been provided or has been provided and set to a value other than 'Policy-Min-TLS-1-2-2019-07'
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'DomainEndpointOptions' has been provided
#     And: 'EnforceHTTPS' in 'DomainEndpointOptions' has been provided and set to bool(true)
#     And: 'TLSSecurityPolicy' in 'DomainEndpointOptions' has not been provided or has been provided and set to a value other than 'Policy-Min-TLS-1-2-2019-07'
#     Then: FAIL
#   Scenario: 5
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'DomainEndpointOptions' has been provided
#     And: 'EnforceHTTPS' in 'DomainEndpointOptions' has been provided and set to bool(true)
#     And: 'TLSSecurityPolicy' in 'DomainEndpointOptions' has been provided and set to 'Policy-Min-TLS-1-2-2019-07'
#     Then: FAIL
#   Scenario: 6
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Elasticsearch domain resource
#     And: 'DomainEndpointOptions' has been provided
#     And: 'EnforceHTTPS' in 'DomainEndpointOptions' has been provided and set to bool(true)
And: 'TLSSecurityPolicy' in 'DomainEndpointOptions' has been provided and set
to 'Policy-Min-TLS-1-2-2019-07'
Then: PASS

# Constants
let ELASTICSEARCH_DOMAIN_TYPE = "AWS::Elasticsearch::Domain"
let INPUT_DOCUMENT = this

# Assignments
let elasticsearch_domains = Resources.*[ Type == %ELASTICSEARCH_DOMAIN_TYPE ]

# Primary Rules
rule elasticsearch_https_required_check when is_cfn_template(%INPUT_DOCUMENT)
%elasticsearch_domains not empty {
  check(%elasticsearch_domains.Properties)
  %elasticsearch_domains not empty {
    [CT.OPENSEARCH.PR.8]: Require an Elasticsearch Service domain to use TLSv1.2
    [FIX]: Within 'DomainEndpointOptions', set 'EnforceHTTPS' to 'true' and set
    'TLSSecurityPolicy' to 'Policy-Min-TLS-1-2-2019-07'.
  }
}

rule elasticsearch_https_required_check when is_cfn_hook(%INPUT_DOCUMENT, %ELASTICSEARCH_DOMAIN_TYPE) {
  check(%INPUT_DOCUMENT.%ELASTICSEARCH_DOMAIN_TYPE.resourceProperties)
  [CT.OPENSEARCH.PR.8]: Require an Elasticsearch Service domain to use TLSv1.2
  [FIX]: Within 'DomainEndpointOptions', set 'EnforceHTTPS' to 'true' and set
  'TLSSecurityPolicy' to 'Policy-Min-TLS-1-2-2019-07'.
}

# Parameterized Rules

rule check(elasticsearch_domain) {
  %elasticsearch_domain {
    # Scenario 2
    DomainEndpointOptions exists
    DomainEndpointOptions is_struct
    DomainEndpointOptions {
      # Scenarios 3, 4, 5 and 6
      EnforceHTTPS exists
      EnforceHTTPS == true
      TLSSecurityPolicy exists
      TLSSecurityPolicy in %ALLOWED_TLS_POLICIES
    }
  }
}

# Utility Rules
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
CT.OPENSEARCH.PR.8 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.elasticsearch
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'
    DomainEndpointOptions:
      EnforceHTTPS: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ElasticsearchDomain:
  Type: AWS::Elasticsearch::Domain
  Properties:
    ElasticsearchVersion: 7.1
    ElasticsearchClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.elasticsearch
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
[CT.OPENSEARCH.PR.9] Require an Amazon OpenSearch Service domain to encrypt data at rest

This control checks whether Amazon OpenSearch Service domains have encryption-at-rest enabled.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::OpenSearchService::Domain
- **AWS CloudFormation guard rule:** [CT.OPENSEARCH.PR.9 rule specification](p. 1025)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.OPENSEARCH.PR.9 rule specification](p. 1025)
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.OPENSEARCH.PR.9 example templates](p. 1027)

**Explanation**

For an added layer of security for sensitive data, you should configure your OpenSearch Service domain to be encrypted at rest. When you configure encryption of data at rest, AWS KMS stores and manages your encryption keys. To perform the encryption, AWS KMS uses the Advanced Encryption Standard algorithm with 256-bit keys (AES-256).

**Remediation for rule failure**

Within EncryptionAtRestOptions, set Enabled to true.

The examples that follow show how to implement this remediation.

**Amazon OpenSearch Service Domain - Example**

An Amazon OpenSearch Service domain configured with encryption-at-rest enabled The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "OpenSearchServiceDomain": {
        "Type": "AWS::OpenSearchService::Domain",
        "Properties": {
            "EngineVersion": "OpenSearch_1.3",
            "ClusterConfig": {
                "InstanceCount": "1",
                "InstanceType": "t3.small.search"
            }
        }
    }
}
```
YAML example

```
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
  Properties:
    EngineVersion: OpenSearch_1.3
    ClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.search
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'
    EncryptionAtRestOptions:
      Enabled: true
```

CT.OPENSEARCH.PR.9 rule specification

```
# #########################################################################
##       Rule Specification       ##
#########################################################################
#
```
# Rule Identifier:
#   opensearch_encrypted_at_rest_check
#
# Description:
#   This control checks whether Amazon OpenSearch Service domains have encryption-at-rest enabled.
#
# Reports on:
#   AWS::OpenSearchService::Domain
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any OpenSearch Service domain resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document contains an OpenSearch Service domain resource
#     And: 'EncryptionAtRestOptions' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document contains an OpenSearch Service domain resource
#     And: 'EncryptionAtRestOptions' has been provided
#     And: In 'EncryptionAtRestOptions', 'Enabled' has not been provided or provided and set to a value other than bool(true)
#     Then: FAIL
#   Scenario: 4
#     Given: The input document contains an OpenSearch Service domain resource
#     And: 'EncryptionAtRestOptions' has been provided
#     And: In 'EncryptionAtRestOptions','Enabled' has been provided and set to bool(true)
#     Then: PASS
#
# Constants
#
# let OPENSEARCH_SERVICE_DOMAIN_TYPE = "AWS::OpenSearchService::Domain"
let INPUT_DOCUMENT = this
#
# Assignments
#
# let opensearch_service_domains = Resources.*[ Type == %OPENSEARCH_SERVICE_DOMAIN_TYPE ]
#
# Primary Rules
#
rule opensearch_encrypted_at_rest_check when is_cfn_template(%INPUT_DOCUMENT)
  %opensearch_service_domains not empty {
    check(%opensearch_service_domains.Properties)
    <<<
      [CT.OPENSEARCH.PR.9]: Require an Amazon OpenSearch Service domain to encrypt data at rest
      [FIX]: Within 'EncryptionAtRestOptions', set 'Enabled' to 'true'.
    >>>
  }
rule opensearch_encrypted_at_rest_check when is_cfn_hook(%INPUT_DOCUMENT, %OPENSEARCH_SERVICE_DOMAIN_TYPE) {
    check(%INPUT_DOCUMENT.%OPENSEARCH_SERVICE_DOMAIN_TYPE.resourceProperties)
    <<
    [CT.OPENSEARCH.PR.9]: Require an Amazon OpenSearch Service domain to encrypt data at rest
    [FIX]: Within 'EncryptionAtRestOptions', set 'Enabled' to 'true'.
    >>
}

# Parameterized Rules
# rule check(opensearch_service_domain) {
%opensearch_service_domain {
    # Scenario 2
    EncryptionAtRestOptions exists
    EncryptionAtRestOptions is_struct

    EncryptionAtRestOptions {
        # Scenarios 3 and 4
        Enabled exists
        Enabled == true
    }
}
}

# Utility Rules
# rule is_cfn_template(doc) {
%doc {
    AWSTemplateFormatVersion exists or
    Resources exists
}
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.OPENSEARCH.PR.9 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  OpenSearchServiceDomain:
    Type: AWS::OpenSearchService::Domain
    Properties:
      EngineVersion: OpenSearch_1.3
      ClusterConfig:
        InstanceCount: '1'
        InstanceType: t3.small.search
      EBSOptions:
        EBSEnabled: true
        Iops: '3000'
        VolumeSize: '10'
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
  Properties:
    EngineVersion: OpenSearch_1.3
    ClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.search
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'
      EncryptionAtRestOptions:
        Enabled: false

[CT.OPENSEARCH.PR.10] Require an Amazon OpenSearch Service domain to be created in a user-specified Amazon VPC

This control checks whether Amazon OpenSearch Service domains are configured with VPC option settings that specify a target Amazon VPC.

• Control objective: Limit network access
• Implementation: AWS CloudFormation guard rule
• Control behavior: Proactive
• Resource types: AWS::OpenSearchService::Domain
• AWS CloudFormation guard rule: CT.OPENSEARCH.PR.10 rule specification (p. 1030)

Details and examples
• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.OPENSEARCH.PR.10 rule specification (p. 1030)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see:
  CT.OPENSEARCH.PR.10 example templates (p. 1032)

Explanation

Ensure that OpenSearch domains are not attached to public subnets. See Resource-based policies in the Amazon OpenSearch Service Developer Guide. Also ensure that your VPC is configured according to the recommended best practices. See Security best practices for your VPC in the Amazon VPC User Guide.

OpenSearch domains deployed within a VPC can communicate with VPC resources over the private AWS network, without the need to traverse the public internet. This configuration increases the security posture by limiting access to the data in transit. VPCs provide a number of network controls to secure access to OpenSearch domains, including network ACL and security groups. AWS Control Tower recommends that you migrate public OpenSearch domains to VPCs to take advantage of these controls.

Remediation for rule failure

Within VPCOptions, set SubnetIds to a list with one or more Amazon EC2 subnet IDs.

The examples that follow show how to implement this remediation.

Amazon OpenSearch Service Domain - Example

An Amazon OpenSearch Service domain configured to deploy within an Amazon VPC by means of VPC option settings. The example is shown in JSON and in YAML.

JSON example

```json
{
  "OpenSearchServiceDomain": {
    "Type": "AWS::OpenSearchService::Domain",
    "Properties": {
      "EngineVersion": "OpenSearch_1.3",
      "ClusterConfig": {
        "InstanceCount": "1",
        "InstanceType": "t3.small.search"
      },
      "EBSOptions": {
        "EBSEnabled": true,
        "Iops": "3000",
        "VolumeSize": "10",
        "VolumeType": "gp3"
      },
      "AccessPolicies": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Deny",
            "Principal": {
              "AWS": "*"
            },
            "Action": "es:*",
            "Resource": "*"
          }
        ],
        "VPCOptions": {
          "SubnetIds": [
            { "Ref": "Subnet" }
          ]
        }
      }
    }
  }
}  ```
YAML example

```
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
Properties:
  EngineVersion: OpenSearch_1.3
  ClusterConfig:
    InstanceCount: '1'
    InstanceType: t3.small.search
  EBSOptions:
    EBSEnabled: true
    Iops: '3000'
    VolumeSize: '10'
    VolumeType: gp3
  AccessPolicies:
    Version: '2012-10-17'
    Statement:
      - Effect: Deny
        Principal:
          AWS: '*'
        Action: es:*
        Resource: '*'
  VPCOptions:
    SubnetIds:
      - !Ref 'Subnet'
```

CT.OPENSEARCH.PR.10 rule specification

```
# ####################################################################
##       Rule Specification        
# ####################################################################
#
# Rule Identifier:
#   opensearch_in_vpc_only_check
#
# Description:
#   This control checks whether Amazon OpenSearch Service domains are configured
#   with VPC option settings that specify a target Amazon VPC.
# # Reports on:
#   AWS::OpenSearchService::Domain
# # Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# # Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
```
# And: The input document does not contain any OpenSearch Service domain resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an OpenSearch Service domain resource
# And: 'VPCOptions' has not been provided
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an OpenSearch Service domain resource
# And: 'VPCOptions' has been provided
# And: 'SubnetIds' in 'VPCOptions' has not been provided or has been provided as an empty list
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an OpenSearch Service domain resource
# And: 'VPCOptions' has been provided
# And: 'SubnetIds' in 'VPCOptions' has been provided as a list with one or more values
# Then: PASS

# Constants
#
let OPENSEARCH_SERVICE_DOMAIN_TYPE = "AWS::OpenSearchService::Domain"
let INPUT_DOCUMENT = this
#
# Assignments
#
let opensearch_service_domains = Resources.*[ Type == %OPENSEARCH_SERVICE_DOMAIN_TYPE ]
#
# Primary Rules
#
rule opensearch_in_vpc_only_check when is_cfn_template(%INPUT_DOCUMENT)
%opensearch_service_domains not empty {
  check(%opensearch_service_domains.Properties)
  <<
  [CT.OPENSEARCH.PR.10]: Require an Amazon OpenSearch Service domain to be created in a user-specified Amazon VPC
  [FIX]: Within 'VPCOptions', set 'SubnetIds' to a list with one or more Amazon EC2 subnet IDs.
  >>
}
rule opensearch_in_vpc_only_check when is_cfn_hook(%INPUT_DOCUMENT, %OPENSEARCH_SERVICE_DOMAIN_TYPE) {
  check(%INPUT_DOCUMENT.%OPENSEARCH_SERVICE_DOMAIN_TYPE.resourceProperties)
  <<
  [CT.OPENSEARCH.PR.10]: Require an Amazon OpenSearch Service domain to be created in a user-specified Amazon VPC
  [FIX]: Within 'VPCOptions', set 'SubnetIds' to a list with one or more Amazon EC2 subnet IDs.
  >>
}
#
# Parameterized Rules
#
rule check(opensearch_service_domain) {
  %opensearch_service_domain {
# Scenario 2
VPCOptions exists
VPCOptions is_struct

VPCOptions {
    # Scenarios 3 and 4
    SubnetIds exists
    SubnetIds is_list
    SubnetIds not empty
}

# Utility Rules
# is_cfn_template(doc) {
%doc {
    AWSTemplateFormatVersion exists or
    Resources exists
}
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.OPENSEARCH.PR.10 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
    Type: 'AWS::EC2::VPC'
    Properties:
        CidrBlock: 10.0.0.0/16
Subnet:
    Type: 'AWS::EC2::Subnet'
    Properties:
        VpcId:
            Ref: VPC
        CidrBlock: 10.0.0.0/16
        AvailabilityZone:
            Fn::Select:
                - 0
                - Fn::GetAZs: ''
OpenSearchServiceDomain:
    Type: AWS::OpenSearchService::Domain
    Properties:
        EngineVersion: OpenSearch_1.3
        ClusterConfig:
            InstanceCount: '1'
            InstanceType: t3.small.search
        EBSOptions:
            EBSEnabled: true
            Iops: '3000'
            VolumeSize: '10'
            VolumeType: gp3
        AccessPolicies:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
Properties:
  EngineVersion: OpenSearch_1.3
  ClusterConfig:
    InstanceCount: '1'
    InstanceType: t3.small.search
  EBSOptions:
    EBSEnabled: true
    Iops: '3000'
    VolumeSize: '10'
    VolumeType: gp3
  AccessPolicies:
    Version: '2012-10-17'
    Statement:
      - Effect: Deny
      - Principal:
        AWS: '*'
      - Action: es:*
      - Resource: '*'

[CT.OPENSEARCH.PR.11] Require an Amazon OpenSearch Service domain to encrypt data sent between nodes

This control checks whether Amazon OpenSearch Service domains have node-to-node encryption enabled.

- Control objective: Encrypt data in transit
- Implementation: AWS CloudFormation guard rule
- Control behavior: Proactive
- Resource types: AWS::OpenSearchService::Domain
- AWS CloudFormation guard rule: CT.OPENSEARCH.PR.11 rule specification (p. 1035)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.OPENSEARCH.PR.11 rule specification (p. 1035)
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: CT.OPENSEARCH.PR.11 example templates (p. 1037)
**Explanation**

HTTPS (TLS) can help prevent potential attackers from eavesdropping on or manipulating network traffic using person-in-the-middle, or similar, attacks. Only encrypted connections over HTTPS (TLS) should be allowed. Enabling node-to-node encryption for OpenSearch domains ensures that intra-cluster communications are encrypted in transit.

**Usage considerations**

- A performance penalty may be associated with this configuration. You should be aware of the performance trade-offs and test them before enabling this option.

**Remediation for rule failure**

Within `NodeToNodeEncryptionOptions`, set `Enabled` to `true`.

The examples that follow show how to implement this remediation.

**Amazon OpenSearch Service Domain - Example**

An Amazon OpenSearch Service domain configured with node-to-node encryption enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "OpenSearchServiceDomain": {
    "Type": "AWS::OpenSearchService::Domain",
    "Properties": {
      "EngineVersion": "OpenSearch_1.3",
      "ClusterConfig": {
        "InstanceCount": "1",
        "InstanceType": "t3.small.search"
      },
      "EBSOptions": {
        "EBSEnabled": true,
        "Iops": "3000",
        "VolumeSize": "10",
        "VolumeType": "gp3"
      },
      "AccessPolicies": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Deny",
            "Principal": {
              "AWS": "*"
            },
            "Action": "es:*",
            "Resource": "*"
          }
        ],
        "NodeToNodeEncryptionOptions": {
          "Enabled": true
        }
      }
    }
  }
}
```

**YAML example**

```yaml
1034
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
  Properties:
    EngineVersion: OpenSearch_1.3
    ClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.search
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'
    NodeToNodeEncryptionOptions:
      Enabled: true

CT.OPENSEARCH.PR.11 rule specification

```
# #################################################################
## Rule Specification       #
# #################################################################

# Rule Identifier:
# opensearch_node_to_node_encryption_check

# Description:
# This control checks whether Amazon OpenSearch Service domains have node-to-node encryption enabled.

# Reports on:
# AWS::OpenSearchService::Domain

# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document does not contain any OpenSearch Service domain resources
#   Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains an OpenSearch Service domain resource
#   And: 'NodeToNodeEncryptionOptions' has not been provided
#   Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
```
And: The input document contains an OpenSearch Service domain resource
And: 'NodeToNodeEncryptionOptions' has been provided
And: In 'NodeToNodeEncryptionOptions', 'Enabled' has not been provided or has been provided and set to a value other than bool(true)
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an OpenSearch Service domain resource
And: 'NodeToNodeEncryptionOptions' has been provided
And: In 'NodeToNodeEncryptionOptions', 'Enabled' has been provided and set to a value of bool(true)
Then: PASS

Constants

let OPENSEARCH_SERVICE_DOMAIN_TYPE = "AWS::OpenSearchService::Domain"
let INPUT_DOCUMENT = this

Assignments

let opensearch_service_domains = Resources.*[ Type == %OPENSEARCH_SERVICE_DOMAIN_TYPE ]

Primary Rules

rule opensearch_node_to_node_encryption_check when is_cfn_template(%INPUT_DOCUMENT)
%opensearch_service_domains not empty {
  check(%opensearch_service_domains.Properties) <<
  [CT.OPENSEARCH.PR.11]: Require an Amazon OpenSearch Service domain to encrypt data sent between nodes
  [FIX]: Within 'NodeToNodeEncryptionOptions', set 'Enabled' to 'true'.
}>>

rule opensearch_node_to_node_encryption_check when is_cfn_hook(%INPUT_DOCUMENT, %OPENSEARCH_SERVICE_DOMAIN_TYPE) {
  check(%INPUT_DOCUMENT.%OPENSEARCH_SERVICE_DOMAIN_TYPE.resourceProperties) <<
  [CT.OPENSEARCH.PR.11]: Require an Amazon OpenSearch Service domain to encrypt data sent between nodes
  [FIX]: Within 'NodeToNodeEncryptionOptions', set 'Enabled' to 'true'.
}>>

Parameterized Rules

rule check(opensearch_service_domain) {
  %opensearch_service_domain {
    # Scenario 2
    NodeToNodeEncryptionOptions exists
    NodeToNodeEncryptionOptions is_struct
    NodeToNodeEncryptionOptions {
      # Scenarios 3 and 4
      Enabled exists
      Enabled == true
    }
  }
}
CT.OPENSEARCH.PR.11 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
  Properties:
    EngineVersion: OpenSearch_1.3
    ClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.search
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal: AWS: '*'
          Action: es:*'
          Resource: '*'
    NodeToNodeEncryptionOptions:
      Enabled: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
  Properties:
    EngineVersion: OpenSearch_1.3
    ClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.search
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
Proactive controls

VolumeSize: '10'
VolumeType: gp3
AccessPolicies:
  Version: '2012-10-17'
  Statement:
    - Effect: Deny
      Principal:
        AWS: '*'
      Action: es:*
      Resource: '*'
    NodeToNodeEncryptionOptions: {}

[CT.OPENSEARCH.PR.12] Require an Amazon OpenSearch Service domain to send error logs to Amazon CloudWatch Logs

This control checks whether Amazon OpenSearch Service domains are configured to send error logs to an Amazon CloudWatch Logs log group.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::OpenSearchService::Domain
- **AWS CloudFormation guard rule:** [CT.OPENSEARCH.PR.12 rule specification (p. 1040)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.OPENSEARCH.PR.12 rule specification (p. 1040)]
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.OPENSEARCH.PR.12 example templates (p. 1043)]

Explanation

Enable error logs (ES_APPLICATION_LOGS) for OpenSearch Service domains and send those logs to Amazon CloudWatch Logs for retention and response. Domain error logs can assist with security and access audits, and can help to diagnose availability issues.

Remediation for rule failure

Within LogPublishingOptions, provide an ES_APPLICATION_LOGS configuration, set Enabled to true, and set CloudWatchLogsLogGroupArn to the ARN of a valid Amazon CloudWatch Logs log group.

The examples that follow show how to implement this remediation.

Amazon OpenSearch Service Domain - Example

An Amazon OpenSearch Service domain configured to send error logs to Amazon CloudWatch Logs. The example is shown in JSON and in YAML.

JSON example

```json
{
```
"OpenSearchServiceDomain": {
  "Type": "AWS::OpenSearchService::Domain",
  "Properties": {
    "EngineVersion": "OpenSearch_1.3",
    "ClusterConfig": {
      "InstanceCount": "1",
      "InstanceType": "t3.small.search"
    },
    "EBSOptions": {
      "EBSEnabled": true,
      "Iops": "3000",
      "VolumeSize": "10",
      "VolumeType": "gp3"
    },
    "AccessPolicies": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Deny",
          "Principal": {
            "AWS": "*"
          },
          "Action": "es:*",
          "Resource": "*"
        }
      ]
    }
  }
},
"LogPublishingOptions": {
  "ES_APPLICATION_LOGS": {
    "CloudWatchLogsLogGroupArn": {
      "Fn::GetAtt": [
        "LogGroup",
        "Arn"
      ],
      "Enabled": true
    }
  }
}

YAML example

OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
Properties:
  EngineVersion: OpenSearch_1.3
  ClusterConfig:
    InstanceCount: '1'
    InstanceType: t3.small.search
  EBSOptions:
    EBSEnabled: true
    Iops: '3000'
    VolumeSize: '10'
    VolumeType: gp3
  AccessPolicies:
    Version: '2012-10-17'
    Statement:
      - Effect: Deny
        Principal:
          AWS: '*'
        Action: es:*
CT.OPENSEARCH.PR.12 rule specification

# #####################################################################
##       Rule Specification        ##
#####################################################################
# Rule Identifier:
#   opensearch_application_logging_enabled_check
#
# Description:
#   This control checks whether Amazon OpenSearch Service domains are configured to send
#   error logs to an Amazon CloudWatch Logs log group.
#
# Reports on:
#   AWS::OpenSearchService::Domain
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any OpenSearch Service domain resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an OpenSearch Service domain resource
#     And: 'LogPublishingOptions' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an OpenSearch Service domain resource
#     And: 'LogPublishingOptions' has been provided
#     And: 'ES_APPLICATION_LOGS' in 'LogPublishingOptions' has not been provided
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an OpenSearch Service domain resource
#     And: 'LogPublishingOptions' has been specified
#     And: 'ES_APPLICATION_LOGS' in 'LogPublishingOptions' has been provided
#     And: 'Enabled' in 'ES_APPLICATION_LOGS' has not been provided or provided and set
to a value other than bool(true)
#     And: 'CloudWatchLogsLogGroupArn' in 'ES_APPLICATION_LOGS' has not been provided or
provided as an empty string or invalid local reference
#     Then: FAIL
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an OpenSearch Service Domain resource
# And: 'LogPublishingOptions' has been specified
# And: 'ES_APPLICATION_LOGS' in 'LogPublishingOptions' has been provided
# And: 'Enabled' in 'ES_APPLICATION_LOGS' has been provided and set to bool(true)
# And: 'CloudWatchLogsLogGroupArn' in 'ES_APPLICATION_LOGS' has not been provided or provided as an empty string or invalid local reference
# Then: FAIL

# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an OpenSearch Service Domain resource
# And: 'LogPublishingOptions' has been specified
# And: 'ES_APPLICATION_LOGS' in 'LogPublishingOptions' has been provided
# And: 'Enabled' in 'ES_APPLICATION_LOGS' has not been provided or provided and set to a value other than bool(true)
# And: 'CloudWatchLogsLogGroupArn' in 'ES_APPLICATION_LOGS' has been provided as a non-empty string or valid local reference
# Then: FAIL

# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an OpenSearch Service Domain resource
# And: 'LogPublishingOptions' has been specified
# And: 'ES_APPLICATION_LOGS' in 'LogPublishingOptions' has been provided
# And: 'Enabled' in 'ES_APPLICATION_LOGS' has been provided and set to bool(true)
# And: 'CloudWatchLogsLogGroupArn' in 'ES_APPLICATION_LOGS' has been provided as a non-empty string or valid local reference
# Then: PASS

# Constants
let OPENSEARCH_SERVICE_DOMAIN_TYPE = "AWS::OpenSearchService::Domain"
let INPUT_DOCUMENT = this

# Assignments
let opensearch_service_domains = Resources.*[ Type == %OPENSEARCH_SERVICE_DOMAIN_TYPE ]

# Primary Rules
rule opensearch_application_logging_enabled_check when is_cfn_template(%INPUT_DOCUMENT) {
    %opensearch_service_domains not empty {
        check(%opensearch_service_domains.Properties)
        [CT.OPENSEARCH.PR.12]: Require an Amazon OpenSearch Service domain to send error logs to Amazon CloudWatch Logs
        [FIX]: Within 'LogPublishingOptions', provide an 'ES_APPLICATION_LOGS' configuration, set 'Enabled' to 'true', and set 'CloudWatchLogsLogGroupArn' to the ARN of a valid Amazon CloudWatch Logs log group.
    }
}

rule opensearch_application_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %OPENSEARCH_SERVICE_DOMAIN_TYPE) {
    check(%INPUT_DOCUMENT.%OPENSEARCH_SERVICE_DOMAIN_TYPE.resourceProperties)
}
[CT.OPENSEARCH.PR.12]: Require an Amazon OpenSearch Service domain to send error logs to Amazon CloudWatch Logs

[Fix]: Within 'LogPublishingOptions', provide an 'ES_APPLICATION_LOGS' configuration, set 'Enabled' to 'true', and set 'CloudWatchLogsLogGroupArn' to the ARN of a valid Amazon CloudWatch Logs log group.

---

# Parameterized Rules

# rule check(opensearch_service_domain) {
  %opensearch_service_domain {
    # Scenario 2
    LogPublishingOptions exists
    LogPublishingOptions is_struct

    LogPublishingOptions {
      # Scenario 3
      ES_APPLICATION_LOGS exists
      ES_APPLICATION_LOGS is_struct

      ES_APPLICATION_LOGS {
        # Scenarios 4, 5, 6 and 7
        Enabled exists
        Enabled == true

        CloudWatchLogsLogGroupArn exists
        check_is_string_and_not_empty(CloudWatchLogsLogGroupArn) or
        check_local_references(%INPUT_DOCUMENT, CloudWatchLogsLogGroupArn, "AWS::Logs::LogGroup")
      }
    }
  }
}

# Utility Rules

# rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

rule check_is_string_and_not_empty(value) {
  %value {
    this is_string
    this != /\A\s*\z/
  }
}

rule check_local_references(doc, reference_properties, referenced_resource_type) {
  %reference_properties {
    'Fn::GetAtt' {
      query_for_resource(%doc, this[0], %referenced_resource_type)
      <<Local Stack reference was invalid>>
    } or Ref {
      query_for_resource(%doc, this, %referenced_resource_type)
      <<Local Stack reference was invalid>>
    }
  }
}
CT.OPENSEARCH.PR.12 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
DependsOn: LogGroupPolicy
Properties:
  EngineVersion: OpenSearch_1.3
  ClusterConfig:
    InstanceCount: '1'
    InstanceType: t3.small.search
EBSEnables:
  EBSEnabled: true
  Iops: '3000'
  VolumeSize: '10'
  VolumeType: gp3
AccessPolicies:
  Version: '2012-10-17'
  Statement:
    - Effect: Deny
      Principal:
        AWS: '*'
      Action: es:*
      Resource: '*'
LogPublishingOptions:
  ES_APPLICATION_LOGS:
    CloudWatchLogsLogGroupArn:
      Fn::GetAtt:
        - LogGroup
        - Arn
      Enabled: true
LogGroup:
  Type: AWS::Logs::LogGroup
LogGroupPolicy:
  Type: AWS::Logs::ResourcePolicy
Properties:
  PolicyName:
    Fn::Sub: ${AWS::StackName}-AllowOS
  PolicyDocument:
    Fn::Sub:
    - LogGroupArn:
      Fn::GetAtt: [LogGroup, Arn]
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

<table>
<thead>
<tr>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenSearchServiceDomain:</td>
</tr>
<tr>
<td>Type: AWS::OpenSearchService::Domain</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>EngineVersion: OpenSearch_1.3</td>
</tr>
<tr>
<td>ClusterConfig:</td>
</tr>
<tr>
<td>InstanceCount: '1'</td>
</tr>
<tr>
<td>InstanceType: t3.small.search</td>
</tr>
<tr>
<td>EBSOptions:</td>
</tr>
<tr>
<td>EBSEnabled: true</td>
</tr>
<tr>
<td>Iops: '3000'</td>
</tr>
<tr>
<td>VolumeSize: '10'</td>
</tr>
<tr>
<td>VolumeType: gp3</td>
</tr>
<tr>
<td>AccessPolicies:</td>
</tr>
<tr>
<td>Version: '2012-10-17'</td>
</tr>
<tr>
<td>Statement:</td>
</tr>
<tr>
<td>- Effect: Deny</td>
</tr>
<tr>
<td>Principal:</td>
</tr>
<tr>
<td>AWS: '*'</td>
</tr>
<tr>
<td>Action: es:*</td>
</tr>
<tr>
<td>Resource: '*'</td>
</tr>
</tbody>
</table>

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

<table>
<thead>
<tr>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenSearchServiceDomain:</td>
</tr>
<tr>
<td>Type: AWS::OpenSearchService::Domain</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>EngineVersion: OpenSearch_1.3</td>
</tr>
<tr>
<td>ClusterConfig:</td>
</tr>
<tr>
<td>InstanceCount: '1'</td>
</tr>
<tr>
<td>InstanceType: t3.small.search</td>
</tr>
<tr>
<td>EBSOptions:</td>
</tr>
<tr>
<td>EBSEnabled: true</td>
</tr>
<tr>
<td>Iops: '3000'</td>
</tr>
<tr>
<td>VolumeSize: '10'</td>
</tr>
<tr>
<td>VolumeType: gp3</td>
</tr>
<tr>
<td>AccessPolicies:</td>
</tr>
<tr>
<td>Version: '2012-10-17'</td>
</tr>
<tr>
<td>Statement:</td>
</tr>
<tr>
<td>- Effect: Deny</td>
</tr>
<tr>
<td>Principal:</td>
</tr>
<tr>
<td>AWS: '*'</td>
</tr>
<tr>
<td>Action: es:*</td>
</tr>
<tr>
<td>Resource: '*'</td>
</tr>
</tbody>
</table>

LogPublishingOptions: |
| ES_APPLICATION_LOGS: |
| Enabled: false |
[CT.OPENSEARCH.PR.13] Require an Amazon OpenSearch Service domain to send audit logs to Amazon CloudWatch Logs

This control checks whether Amazon OpenSearch Service domains are configured to send audit logs to an Amazon CloudWatch Logs log group.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::OpenSearchService::Domain
- **AWS CloudFormation guard rule:** [CT.OPENSEARCH.PR.13 rule specification (p. 1047)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.OPENSEARCH.PR.13 rule specification (p. 1047)]
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.OPENSEARCH.PR.13 example templates (p. 1050)]

**Explanation**

Audit logs are highly customizable. They allow you to track user activity on your OpenSearch clusters, including authentication successes and failures, requests to OpenSearch, index changes, and incoming search queries.

**Usage considerations**

- Audit log publishing requires advanced security options to be enabled on Amazon OpenSearch Service domains.
- To enable advanced security options on an Amazon OpenSearch Service domain, you must enable encryption of data at rest by means of the EncryptionAtRestOptions property, node-to-node encryption by means of the NodeToNodeEncryptionOptions property, and enforce HTTPS connections by means of the EnforceHTTPS property within DomainEndpointOptions.

**Remediation for rule failure**

Within LogPublishingOptions, provide an AUDIT_LOGS configuration, set Enabled to true and CloudWatchLogsLogGroupArn to the ARN of a valid Amazon CloudWatch Logs log group.

The examples that follow show how to implement this remediation.

**Amazon OpenSearch Service Domain - Example**

An Amazon OpenSearch Service domain configured to send audit logs to Amazon CloudWatch Logs. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "OpenSearchServiceDomain": {
        "Type": "AWS::OpenSearchService::Domain",
        "DependsOn": "LogGroupPolicy",
        "Properties": {
            "EngineVersion": "OpenSearch_1.3",
```

[1045]
"ClusterConfig": {
  "InstanceCount": "1",
  "InstanceType": "t3.small.search"
},
"EBSOptions": {
  "EBSEnabled": true,
  "Iops": "3000",
  "VolumeSize": "10",
  "VolumeType": "gp3"
},
"AccessPolicies": {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Resource": "*"
    }
  ]
},
"NodeToNodeEncryptionOptions": {
  "Enabled": true
},
"EncryptionAtRestOptions": {
  "Enabled": true
},
"DomainEndpointOptions": {
  "EnforceHTTPS": true
},
"AdvancedSecurityOptions": {
  "Enabled": true,
  "InternalUserDatabaseEnabled": false,
  "MasterUserOptions": {
    "MasterUserARN": {
      "Fn::GetAtt": [
        "IAMRole",
        "Arn"
      ]
    }
  }
},
"LogPublishingOptions": {
  "AUDIT_LOGS": {
    "CloudWatchLogsLogGroupArn": {
      "Fn::GetAtt": [
        "LogGroup",
        "Arn"
      ],
      "Enabled": true
    }
  }
}
AWS Control Tower User Guide
Proactive controls

DependsOn: LogGroupPolicy
Properties:
  EngineVersion: OpenSearch_1.3
  ClusterConfig:
    InstanceCount: '1'
    InstanceType: t3.small.search
  EBSOptions:
    EBSEnabled: true
    Iops: '3000'
    VolumeSize: '10'
    VolumeType: gp3
  AccessPolicies:
    Version: '2012-10-17'
    Statement:
      - Effect: Deny
        Principal:
          AWS: '*'
        Action: es:*
        Resource: '*'
  NodeToNodeEncryptionOptions:
    Enabled: true
  EncryptionAtRestOptions:
    Enabled: true
  DomainEndpointOptions:
    EnforceHTTPS: true
  AdvancedSecurityOptions:
    Enabled: true
    InternalUserDatabaseEnabled: false
    MasterUserOptions:
      MasterUserARN: !GetAtt 'IAMRole.Arn'
  LogPublishingOptions:
    AUDIT_LOGS:
      CloudWatchLogsLogGroupArn: !GetAtt 'LogGroup.Arn'
      Enabled: true

---

CT.OPENSEARCH.PR.13 rule specification

# ****************************
#  Rule Specification       #
#  ****************************
#
# Rule Identifier:
# opensearch_audit_logging_enabled_check
#
# Description:
# This control checks whether Amazon OpenSearch Service domains are configured to send
# audit logs to an Amazon CloudWatch Logs log group.
#
# Reports on:
# AWS::OpenSearchService::Domain
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document does not contain any OpenSearch Service domain resources
Then: SKIP

Scenario: 2
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
   And: The input document contains an OpenSearch Service domain resource
   And: 'LogPublishingOptions' has not been provided
Then: FAIL

Scenario: 3
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
   And: The input document contains an OpenSearch Service domain resource
   And: 'LogPublishingOptions' has been provided
   And: 'AUDIT_LOGS' in 'LogPublishingOptions' has not been provided
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
   And: The input document contains an OpenSearch Service domain resource
   And: 'LogPublishingOptions' has been specified
   And: 'AUDIT_LOGS' in 'LogPublishingOptions' has been provided
   And: 'Enabled' in 'AUDIT_LOGS' has not been provided or provided and set to
   a value other than bool(true)
   And: 'CloudWatchLogsLogGroupArn' in 'AUDIT_LOGS' has not been provided or provided
   as an empty string or invalid local reference
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
   And: The input document contains an OpenSearch Service domain resource
   And: 'LogPublishingOptions' has been specified
   And: 'AUDIT_LOGS' in 'LogPublishingOptions' has been provided
   And: 'Enabled' in 'AUDIT_LOGS' has been provided and set to bool(true)
   And: 'CloudWatchLogsLogGroupArn' in 'AUDIT_LOGS' has not been provided or provided
   as an empty string or invalid local reference
Then: FAIL

Scenario: 6
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
   And: The input document contains an OpenSearch Service domain resource
   And: 'LogPublishingOptions' has been specified
   And: 'AUDIT_LOGS' in 'LogPublishingOptions' has been provided
   And: 'Enabled' in 'AUDIT_LOGS' has been provided and set to bool(true)
   And: 'CloudWatchLogsLogGroupArn' in 'AUDIT_LOGS' has been provided as a non-empty
   string
   or valid local reference
Then: FAIL

Scenario: 7
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
   And: The input document contains an OpenSearch Service domain resource
   And: 'LogPublishingOptions' has been specified
   And: 'AUDIT_LOGS' in 'LogPublishingOptions' has been provided
   And: 'Enabled' in 'AUDIT_LOGS' has been provided and set to bool(true)
   And: 'CloudWatchLogsLogGroupArn' in 'AUDIT_LOGS' has been provided as a non-empty
   string
   or valid local reference
Then: PASS

# Constants

let OPENSEARCH_SERVICE_DOMAIN_TYPE = "AWS::OpenSearchService::Domain"
let INPUT_DOCUMENT = this
# Assignments

let opensearch_service_domains = Resources.*[ Type == %OPENSEARCH_SERVICE_DOMAIN_TYPE ]

# Primary Rules

rule opensearchAuditLoggingEnabledCheck when is_cfn_template(%INPUT_DOCUMENT)
    %opensearch_service_domains not empty {
    check(%opensearch_service_domains.Properties)
    <<
    [CT.OPENSEARCH.PR.13]: Require an Amazon OpenSearch Service domain to send audit
    logs to Amazon CloudWatch Logs
    [FIX]: Within 'LogPublishingOptions', provide an 'AUDIT_LOGS' configuration,
    set 'Enabled' to 'true' and 'CloudWatchLogsLogGroupArn' to the ARN of a valid Amazon
    CloudWatch Logs log group.
    >>}

rule opensearchAuditLoggingEnabledCheck when is_cfn_hook(%INPUT_DOCUMENT, %OPENSEARCH_SERVICE_DOMAIN_TYPE) {
    check(%INPUT_DOCUMENT.%OPENSEARCH_SERVICE_DOMAIN_TYPE.resourceProperties)
    <<
    [CT.OPENSEARCH.PR.13]: Require an Amazon OpenSearch Service domain to send audit
    logs to Amazon CloudWatch Logs
    [FIX]: Within 'LogPublishingOptions', provide an 'AUDIT_LOGS' configuration,
    set 'Enabled' to 'true' and 'CloudWatchLogsLogGroupArn' to the ARN of a valid Amazon
    CloudWatch Logs log group.
    >>}

# Parameterized Rules

rule check(opensearch_service_domain) {
    %opensearch_service_domain {
    # Scenario 2
    LogPublishingOptions exists
    LogPublishingOptions is_struct
    LogPublishingOptions {
    # Scenario 3
    AUDIT_LOGS exists
    AUDIT_LOGS is_struct
    AUDIT_LOGS {
    # Scenarios 4, 5, 6 and 7
    Enabled exists
    Enabled == true
    CloudWatchLogsLogGroupArn exists
    check_is_string_and_not_empty(CloudWatchLogsLogGroupArn) or
    check_local_references(%INPUT_DOCUMENT, CloudWatchLogsLogGroupArn,
    "AWS::Logs::LogGroup")
    }
    }
    }
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
    AWSTemplateFormatVersion exists  or
rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

rule check_is_string_and_not_empty(value) {
    %value {
        this is_string
        this != /\A\s*\z/
    }
}

rule check_local_references(doc, reference_properties, referenced_resource_type) {
    %reference_properties {
        'Fn::GetAtt' {
            query_for_resource(%doc, this[0], %referenced_resource_type)
            "<Local Stack reference was invalid>"
        } or Ref {
            query_for_resource(%doc, this, %referenced_resource_type)
            "<Local Stack reference was invalid>"
        }
    }
}

rule query_for_resource(doc, resource_key, referenced_resource_type) {
    let referenced_resource = %doc.Resources[ keys == %resource_key ]
    %referenced_resource not empty
    %referenced_resource {
        Type == %referenced_resource_type
    }
}

CT.OPENSEARCH.PR.13 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
IAMRole:
    Type: AWS::IAM::Role
Properties:
    AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
            - Effect: Allow
              Principal:
                AWS:
                    Ref: AWS::AccountId
              Action: sts:AssumeRole
OpenSearchServiceDomain:
    Type: AWS::OpenSearchService::Domain
DependsOn: LogGroupPolicy
Properties:
    EngineVersion: OpenSearch_1.3
    ClusterConfig:
        InstanceCount: '1'
        InstanceType: t3.small.search
EBSOptions:
  EBSEnabled: true
  Iops: '3000'
  VolumeSize: '10'
  VolumeType: gp3
AccessPolicies:
  Version: '2012-10-17'
  Statement:
    - Effect: Deny
      Principal:
        AWS: '*'
      Action: es:*
      Resource: '*'
NodeToNodeEncryptionOptions:
  Enabled: true
EncryptionAtRestOptions:
  Enabled: true
DomainEndpointOptions:
  EnforceHTTPS: true
AdvancedSecurityOptions:
  Enabled: true
  InternalUserDatabaseEnabled: false
  MasterUserOptions:
    MasterUserARN:
      Fn::GetAtt:
        - IAMRole
        - Arn
LogPublishingOptions:
  AUDIT_LOGS:
    CloudWatchLogsLogGroupArn:
      Fn::GetAtt:
        - LogGroup
        - Arn
    Enabled: true
LogGroup:
  Type: AWS::Logs::LogGroup
LogGroupPolicy:
  Type: AWS::Logs::ResourcePolicy
  Properties:
    PolicyName: AllowES
    PolicyDocument:
      Fn::Sub:
        - LogGroupArn:
          Fn::GetAtt: [ LogGroup, Arn ]

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
IAMRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            AWS:
              Ref: AWS::AccountId
Action: sts:AssumeRole
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
  Properties:
    EngineVersion: OpenSearch_1.3
    ClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.search
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'
    NodeToNodeEncryptionOptions:
      Enabled: true
    EncryptionAtRestOptions:
      Enabled: true
    DomainEndpointOptions:
      EnforceHTTPS: true
    AdvancedSecurityOptions:
      Enabled: true
    InternalUserDatabaseEnabled: false
    MasterUserOptions:
      MasterUserARN:
        Fn::GetAtt:
          - IAMRole
          - Arn

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  IAMRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              AWS:
                Ref: AWS::AccountId
            Action: sts:AssumeRole
  OpenSearchServiceDomain:
    Type: AWS::OpenSearchService::Domain
    Properties:
      EngineVersion: OpenSearch_1.3
      ClusterConfig:
        InstanceCount: '1'
        InstanceType: t3.small.search
      EBSOptions:
        EBSEnabled: true
        Iops: '3000'
        VolumeSize: '10'
        VolumeType: gp3
**AccessPolicies:**
- **Version:** '2012-10-17'
  - **Statement:**
    - **Effect:** Deny
    - **Principal:** AWS: '*'
    - **Action:** es:*
    - **Resource:** '*'

**NodeToNodeEncryptionOptions:**
- **Enabled:** true

**EncryptionAtRestOptions:**
- **Enabled:** true

**DomainEndpointOptions:**
- **EnforceHTTPS:** true

**AdvancedSecurityOptions:**
- **Enabled:** true
  - **InternalUserDatabaseEnabled:** false
  - **MasterUserOptions:**
    - **MasterUserARN:**
      - Fn::GetAtt:
        - IAMRole
        - Arn

**LogPublishingOptions:**
- **AUDIT_LOGS:**
  - **Enabled:** false

---

**[CT.OPENSEARCH.PR.14] Require an Amazon OpenSearch Service domain to have zone awareness and at least three data nodes**

This control checks whether Amazon OpenSearch Service domains are configured with at least three data nodes and zone awareness enabled.

- **Control objective:** Improve availability
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::OpenSearchService::Domain
- **AWS CloudFormation guard rule:** [CT.OPENSEARCH.PR.14 rule specification (p. 1055)]

**Details and examples**
- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.OPENSEARCH.PR.14 rule specification (p. 1055)]
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.OPENSEARCH.PR.14 example templates (p. 1057)]

**Explanation**
An OpenSearch domain requires at least three data nodes for high availability and fault-tolerance. Deploying an OpenSearch domain with at least three data nodes ensures that the cluster can remain operative if a node fails.

**Remediation for rule failure**
Within ClusterConfig, set ZoneAwarenessEnabled to true and InstanceCount to an integer value greater than or equal to three.

The examples that follow show how to implement this remediation.
Amazon OpenSearch Service Domain - Example

An Amazon OpenSearch Service domain configured with three data nodes and zone awareness enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "OpenSearchServiceDomain": {
        "Type": "AWS::OpenSearchService::Domain",
        "Properties": {
            "EngineVersion": "OpenSearch_1.3",
            "EBSOptions": {
                "EBSEnabled": true,
                "Iops": "3000",
                "VolumeSize": "10",
                "VolumeType": "gp3"
            },
            "AccessPolicies": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Deny",
                        "Principal": {
                            "AWS": "*"
                        },
                        "Action": "es:*",
                        "Resource": "*"
                    }
                ]
            },
            "ClusterConfig": {
                "InstanceType": "t3.small.search",
                "InstanceCount": 3,
                "ZoneAwarenessEnabled": true,
                "ZoneAwarenessConfig": {
                    "AvailabilityZoneCount": 3
                }
            }
        }
    }
}
```

**YAML example**

```yaml
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
  Properties:
    EngineVersion: OpenSearch_1.3
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*```
Resource: '*'
ClusterConfig:
  InstanceType: t3.small.search
  InstanceCount: 3
  ZoneAwarenessEnabled: true
  ZoneAwarenessConfig:
    AvailabilityZoneCount: 3

CT.OPENSEARCH.PR.14 rule specification

```bash
# ###################################
##       Rule Specification        ##
#####################################
#
# Rule Identifier:
#   opensearch_data_node_fault_tolerance_check
#
# Description:
#   This control checks whether Amazon OpenSearch Service domains are configured with at least three data nodes and zone awareness enabled.
#
# Reports on:
#   AWS::OpenSearchService::Domain
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document does not contain any OpenSearch Service domain resources
#      Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document contains an OpenSearch Service domain resource
#       And: 'ClusterConfig' has not been provided
#      Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document contains an OpenSearch Service domain resource
#       And: 'ClusterConfig' has been provided
#       And: 'ZoneAwarenessEnabled' in 'ClusterConfig' has not been provided or provided and set to a value other than bool(true)
#       And: 'InstanceCount' in 'ClusterConfig' has not been provided or provided and set to an integer value less than three (< 3)
#      Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document contains an OpenSearch Service domain resource
#       And: 'ClusterConfig' has been provided
#       And: 'ZoneAwarenessEnabled' in 'ClusterConfig' has been provided
#       And: 'InstanceCount' in 'ClusterConfig' has not been provided or provided and set to an integer value less than three (< 3)
```
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an OpenSearch Service domain resource
# And: 'ClusterConfig' has been provided
# And: 'ZoneAwarenessEnabled' in 'ClusterConfig' has not been provided or provided and set to a value other than bool(true)
# And: 'InstanceCount' in 'ClusterConfig' has been provided and set to an integer value greater than or equal to three (>= 3)
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an OpenSearch Service domain resource
# And: 'ClusterConfig' has been provided
# And: 'ZoneAwarenessEnabled' in 'ClusterConfig' has been provided and set to bool(true)
# And: 'InstanceCount' in 'ClusterConfig' has been provided and set to an integer value greater than or equal to three (>= 3)
# Then: PASS

# Constants
# let OPENSEARCH_SERVICE_DOMAIN_TYPE = "AWS::OpenSearchService::Domain"
let INPUT_DOCUMENT = this
#
# Assignments
# let opensearch_service_domains = Resources.*[ Type == %OPENSEARCH_SERVICE_DOMAIN_TYPE ]
#
# Primary Rules
# rule opensearch_data_node_fault_tolerance_check when is_cfn_template(%INPUT_DOCUMENT)
#   %opensearch_service_domains not empty
#   { check(%opensearch_service_domains.Properties)
#     <<
#       [CT.OPENSEARCH.PR.14]: Require an Amazon OpenSearch Service domain to have zone awareness and at least three data nodes
#       [FIX]: Within 'ClusterConfig', set 'ZoneAwarenessEnabled' to 'true' and 'InstanceCount' to an integer value greater than or equal to three.
#     >>
#   }
rule opensearch_data_node_fault_tolerance_check when is_cfn_hook(%INPUT_DOCUMENT, %OPENSEARCH_SERVICE_DOMAIN_TYPE) {
  check(%INPUT_DOCUMENT.%OPENSEARCH_SERVICE_DOMAIN_TYPE.resourceProperties)
  <<
    [CT.OPENSEARCH.PR.14]: Require an Amazon OpenSearch Service domain to have zone awareness and at least three data nodes
    [FIX]: Within 'ClusterConfig', set 'ZoneAwarenessEnabled' to 'true' and 'InstanceCount' to an integer value greater than or equal to three.
  >>
}
#
# Parameterized Rules
# rule check(opensearch_service_domain) {
  %opensearch_service_domain {
    # Scenario 2
    ClusterConfig exists
ClusterConfig is_struct

ClusterConfig {
    # Scenario 3, 4, 5 and 6
    ZoneAwarenessEnabled exists
    ZoneAwarenessEnabled == true
        InstanceCount exists
        InstanceCount >= 3
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.OPENSEARCH.PR.14 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
OpenSearchServiceDomain:
    Type: AWS::OpenSearchService::Domain
Properties:
    EngineVersion: OpenSearch_1.3
    EBSOptions:
        EBSEnabled: true
        Iops: '3000'
        VolumeSize: '10'
        VolumeType: gp3
    AccessPolicies:
        Version: '2012-10-17'
        Statement:
            - Effect: Deny
              Principal:
                  AWS: '*'
              Action: es:*
              Resource: '*'
    ClusterConfig:
        InstanceType: t3.small.search
        InstanceCount: 3
        ZoneAwarenessEnabled: true
        ZoneAwarenessConfig:
            AvailabilityZoneCount: 3

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.
Resources:

OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
  Properties:
    EngineVersion: OpenSearch_1.3
    ClusterConfig:
      InstanceType: t3.small.search
      ZoneAwarenessEnabled: false
      InstanceCount: 2
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'

[CT.OPENSEARCH.PR.15] Require an Amazon OpenSearch Service domain to use fine-grained access control

This control checks whether Amazon OpenSearch Service domains have fine-grained access control enabled.

- **Control objective:** Enforce least privilege
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::OpenSearchService::Domain
- **AWS CloudFormation guard rule:** [CT.OPENSEARCH.PR.15 rule specification](p. 1060)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.OPENSEARCH.PR.15 rule specification](p. 1060)
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.OPENSEARCH.PR.15 example templates](p. 1063)

Explanation

Fine-grained access control offers additional ways of controlling access to your data on Amazon OpenSearch Service.

**Usage considerations**

- Fine-grained access control requires that advanced security options must be enabled on Amazon OpenSearch Service domains.
- To enable advanced security options on an Amazon OpenSearch Service domain, you must enable encryption of data at rest by means of the EncryptionAtRestOptions property, node-to-node encryption by means of the NodeToNodeEncryptionOptions
property, and enforce HTTPS connections by means of the `EnforceHTTPS` property within `DomainEndpointOptions`.

Remediation for rule failure

Within `AdvancedSecurityOptions`, set `Enabled` to `true`, set `InternalUserDatabaseEnabled` to `true` or `false`, and set `MasterUserOptions` with an options configuration for your master user.

The examples that follow show how to implement this remediation.

Amazon OpenSearch Service Domain - Example

An Amazon OpenSearch Service domain configured with fine-grained access control. The example is shown in JSON and in YAML.

JSON example

```json
{
    "OpenSearchServiceDomain": {
        "Type": "AWS::OpenSearchService::Domain",
        "Properties": {
            "EngineVersion": "OpenSearch_1.3",
            "ClusterConfig": {
                "InstanceCount": "1",
                "InstanceType": "t3.small.search"
            },
            "EBSOptions": {
                "EBSEnabled": true,
                "Iops": "3000",
                "VolumeSize": "10",
                "VolumeType": "gp3"
            },
            "AccessPolicies": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Deny",
                        "Principal": {
                            "AWS": "*"
                        },
                        "Action": "es:*",
                        "Resource": "*"
                    }
                ]
            },
            "NodeToNodeEncryptionOptions": {
                "Enabled": true
            },
            "EncryptionAtRestOptions": {
                "Enabled": true
            },
            "DomainEndpointOptions": {
                "EnforceHTTPS": true
            },
            "AdvancedSecurityOptions": {
                "Enabled": true,
                "InternalUserDatabaseEnabled": false,
                "MasterUserOptions": {
                    "MasterUserARN": {
                        "Fn::GetAtt": [
                            "IAMRole",
                            "Arn"
                        ]
                    }
                }
            }
        }
    }
}
```
YAML example

```
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
  Properties:
    EngineVersion: OpenSearch_1.3
    ClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.search
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'
    NodeToNodeEncryptionOptions:
      Enabled: true
    EncryptionAtRestOptions:
      Enabled: true
    DomainEndpointOptions:
      EnforceHTTPS: true
    AdvancedSecurityOptions:
      Enabled: true
    InternalUserDatabaseEnabled: false
    MasterUserOptions:
      MasterUserARN: !GetAtt 'IAMRole.Arn'
```

CT.OPENSEARCH.PR.15 rule specification

```
# ###############################################################
# Rule Specification  
# ########################################################################
# Rule Identifier:        
# opensearch_fine_grained_access_control_enabled_check
# Description:
#   This control checks whether Amazon OpenSearch Service domains have fine-grained access control enabled.
# Reports on:        
#   AWS::OpenSearchService::Domain
```
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any OpenSearch Service domain resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an OpenSearch Service domain resource
#     And: 'AdvancedSecurityOptions' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an OpenSearch Service domain resource
#     And: 'AdvancedSecurityOptions' has been provided
#     And: 'Enabled' in 'AdvancedSecurityOptions' has not been provided or
#          has been provided and set to a value other than bool(true)
#     And: 'InternalUserDatabaseEnabled' in 'AdvancedSecurityOptions' has not been
#          provided or provided and set to a non boolean value
#     And: 'MasterUserOptions' in 'AdvancedSecurityOptions' has not been provided or
#          provided and set to a value other than a struct
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an OpenSearch Service domain resource
#     And: 'AdvancedSecurityOptions' has been provided
#     And: 'Enabled' in 'AdvancedSecurityOptions' has been provided and set to bool(true)
#     And: 'InternalUserDatabaseEnabled' in 'AdvancedSecurityOptions' has not been
#          provided or provided and set to a boolean value
#     And: 'MasterUserOptions' in 'AdvancedSecurityOptions' has not been provided or
#          provided and set to a value other than a struct
#     Then: FAIL
#   Scenario: 5
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an OpenSearch Service domain resource
#     And: 'AdvancedSecurityOptions' has been provided
#     And: 'Enabled' in 'AdvancedSecurityOptions' has been provided and set to bool(true)
#     And: 'InternalUserDatabaseEnabled' in 'AdvancedSecurityOptions' has been provided
#          and set to a boolean value
#     And: 'MasterUserOptions' in 'AdvancedSecurityOptions' has not been provided or
#          provided and set to a value other than a struct
#     Then: FAIL
#   Scenario: 6
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an OpenSearch Service domain resource
#     And: 'AdvancedSecurityOptions' has been provided
#     And: 'Enabled' in 'AdvancedSecurityOptions' has been provided and set to bool(true)
#     And: 'InternalUserDatabaseEnabled' in 'AdvancedSecurityOptions' has not been
#          provided or provided and set to a
Proactive controls

# Given: The input document contains an OpenSearch Service domain resource
# And: 'AdvancedSecurityOptions' has been provided
# And: 'Enabled' in 'AdvancedSecurityOptions' has been provided and set to bool(true)
# And: 'InternalUserDatabaseEnabled' in 'AdvancedSecurityOptions' has been provided and set to a boolean value
# Then: PASS

# Constants
let OPENSEARCH_SERVICE_DOMAIN_TYPE = "AWS::OpenSearchService::Domain"

# Assignments
let opensearch_service_domains = Resources.*[ Type == %OPENSEARCH_SERVICE_DOMAIN_TYPE ]

# Primary Rules
rule opensearch_fine_grained_access_control_enabled_check when is_cfn_template(%INPUT_DOCUMENT) %opensearch_service_domains not empty {
  check(%opensearch_service_domains.Properties)
  
  [CT.OPENSEARCH.PR.15]: Require an Amazon OpenSearch Service domain to use fine-grained access control
  [FIX]: Within 'AdvancedSecurityOptions', set 'Enabled' to 'true', set 'InternalUserDatabaseEnabled' to 'true' or 'false', and set 'MasterUserOptions' with an options configuration for your master user.
}

rule opensearch_fine_grained_access_control_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %OPENSEARCH_SERVICE_DOMAIN_TYPE) {
  check(%INPUT_DOCUMENT.%OPENSEARCH_SERVICE_DOMAIN_TYPE.resourceProperties)
  
  [CT.OPENSEARCH.PR.15]: Require an Amazon OpenSearch Service domain to use fine-grained access control
  [FIX]: Within 'AdvancedSecurityOptions', set 'Enabled' to 'true', set 'InternalUserDatabaseEnabled' to 'true' or 'false', and set 'MasterUserOptions' with an options configuration for your master user.
}

# Parameterized Rules
rule check(opensearch_service_domain) {
  %opensearch_service_domain {
    # Scenario 2
    AdvancedSecurityOptions exists
    AdvancedSecurityOptions is_struct
  }
}
AdvancedSecurityOptions {
    # Scenarios 3, 4, 5, 6 and 7
    Enabled exists
    Enabled == true

    InternalUserDatabaseEnabled exists
    InternalUserDatabaseEnabled in [ true, false ]

    MasterUserOptions exists
    MasterUserOptions is_struct
}
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

**CT.OPENSEARCH.PR.15 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

<table>
<thead>
<tr>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAMRole:</td>
</tr>
<tr>
<td>Type: AWS::IAM::Role</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>AssumeRolePolicyDocument:</td>
</tr>
<tr>
<td>Version: '2012-10-17'</td>
</tr>
<tr>
<td>Statement:</td>
</tr>
<tr>
<td>- Effect: Allow</td>
</tr>
<tr>
<td>Principal:</td>
</tr>
<tr>
<td>AWS:</td>
</tr>
<tr>
<td>Ref: AWS::AccountId</td>
</tr>
<tr>
<td>Action:</td>
</tr>
<tr>
<td>- sts:AssumeRole</td>
</tr>
<tr>
<td>OpenSearchServiceDomain:</td>
</tr>
<tr>
<td>Type: AWS::OpenSearchService::Domain</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>EngineVersion: OpenSearch_1.3</td>
</tr>
<tr>
<td>ClusterConfig:</td>
</tr>
<tr>
<td>InstanceCount: '1'</td>
</tr>
<tr>
<td>InstanceType: t3.small.search</td>
</tr>
<tr>
<td>EBSEOptions:</td>
</tr>
<tr>
<td>EBSEnabled: true</td>
</tr>
<tr>
<td>Iops: '3000'</td>
</tr>
<tr>
<td>VolumeSize: '10'</td>
</tr>
<tr>
<td>VolumeType: gp3</td>
</tr>
<tr>
<td>AccessPolicies:</td>
</tr>
<tr>
<td>Version: '2012-10-17'</td>
</tr>
</tbody>
</table>
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
  Properties:
    EngineVersion: OpenSearch_1.3
    ClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.search
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
    Statement:
      - Effect: Deny
      Principal:
        AWS: '*'
      Action: es:*
      Resource: '*'
    NodeToNodeEncryptionOptions:
      Enabled: true
    EncryptionAtRestOptions:
      Enabled: true
    DomainEndpointOptions:
      EnforceHTTPS: true
    AdvancedSecurityOptions:
      Enabled: false
    MasterUserOptions:
      MasterUserARN:
        Fn::GetAtt: [IAMRole, Arn]

[CT.OPENSEARCH.PR.16] Require an Amazon OpenSearch Service domain to use TLSv1.2

This control checks whether Amazon OpenSearch Service domains are configured to require HTTPS with a minimum TLS version of TLSv1.2.

- **Control objective**: Encrypt data in transit
• **Implementation**: AWS CloudFormation guard rule
• **Control behavior**: Proactive
• **Resource types**: AWS::OpenSearchService::Domain
• **AWS CloudFormation guard rule**: CT.OPENSEARCH.PR.16 rule specification (p. 1066)

**Details and examples**

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.OPENSEARCH.PR.16 rule specification (p. 1066)
• For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: CT.OPENSEARCH.PR.16 example templates (p. 1068)

**Explanation**

HTTPS (TLS) can help prevent potential attackers from using person-in-the-middle, or similar attacks, to eavesdrop on or manipulate network traffic. Only encrypted connections over HTTPS (TLS) should be allowed. Encrypting data in transit can affect performance. You should test your application with this feature to understand the performance profile and the effects of TLS. TLS 1.2 provides several security enhancements over previous versions of TLS.

**Remediation for rule failure**


The examples that follow show how to implement this remediation.

**Amazon OpenSearch Service Domain - Example**

An Amazon OpenSearch Service domain configured to require all traffic to the domain arrive over HTTPS with a minimum TLS version of TLSv1.2. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "OpenSearchServiceDomain": {
        "Type": "AWS::OpenSearchService::Domain",
        "Properties": {
            "EngineVersion": "OpenSearch_1.3",
            "ClusterConfig": {
                "InstanceCount": "1",
                "InstanceType": "t3.small.search"
            },
            "EBSOptions": {
                "EBSEnabled": true,
                "Iops": "3000",
                "VolumeSize": "10",
                "VolumeType": "gp3"
            },
            "AccessPolicies": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Deny",
                        "Principal": {
                            "AWS": "*"
                        }
                    }
                ]
            }
        }
    }
}
```
YAML example

OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
  Properties:
    EngineVersion: OpenSearch_1.3
    ClusterConfig:
      InstanceCount: '1'
      InstanceType: t3.small.search
    EBSOptions:
      EBSEnabled: true
      Iops: '3000'
      VolumeSize: '10'
      VolumeType: gp3
    AccessPolicies:
      Version: '2012-10-17'
      Statement:
        - Effect: Deny
          Principal:
            AWS: '*'
          Action: es:*
          Resource: '*'
    DomainEndpointOptions:
      EnforceHTTPS: true

CT.OPENSEARCH.PR.16 rule specification

# #################################################################################################
##       Rule Specification                        ##
# #  Rule Identifier:                                #
#  opensearch_https_required_check                   #
# #  Description:                                   #
#  This control checks whether Amazon OpenSearch Service domains are configured to require HTTPS with a minimum TLS version of TLSv1.2. #
# #  Reports on:                                    #
#  AWS::OpenSearchService::Domain                   #
# #  Evaluates:                                     #
#  AWS CloudFormation, AWS CloudFormation hook      #
#################################################################################################
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document does not contain any OpenSearch Service domain resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an OpenSearch Service domain resource
# And: 'DomainEndpointOptions' has not been provided
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an OpenSearch Service domain resource
# And: 'DomainEndpointOptions' has been provided
# And: 'EnforceHTTPS' in 'DomainEndpointOptions' has not been provided or
# has been provided and set to a value other than bool(true)
# And: 'TLSSecurityPolicy' in 'DomainEndpointOptions' has not been provided or
# has been provided and set to a value other than 'Policy-Min-TLS-1-2-2019-07'
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an OpenSearch Service domain resource
# And: 'DomainEndpointOptions' has been provided
# And: 'EnforceHTTPS' in 'DomainEndpointOptions' has been provided and set to
bool(true)
# And: 'TLSSecurityPolicy' in 'DomainEndpointOptions' has not been provided or
# has been provided and set to a value other than 'Policy-Min-TLS-1-2-2019-07'
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an OpenSearch Service domain resource
# And: 'DomainEndpointOptions' has been provided
# And: 'EnforceHTTPS' in 'DomainEndpointOptions' has not been provided or
# has been provided and set to a value other than bool(true)
# And: 'TLSSecurityPolicy' in 'DomainEndpointOptions' has been provided and set
to 'Policy-Min-TLS-1-2-2019-07'
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an OpenSearch Service domain resource
# And: 'DomainEndpointOptions' has been provided
# And: 'EnforceHTTPS' in 'DomainEndpointOptions' has been provided and set to
bool(true)
# And: 'TLSSecurityPolicy' in 'DomainEndpointOptions' has been provided and set
to 'Policy-Min-TLS-1-2-2019-07'
# Then: PASS
#
# Constants
#
let OPENSEARCH_SERVICE_DOMAIN_TYPE = "AWS::OpenSearchService::Domain"
let INPUT_DOCUMENT = this
#
# Assignments
let opensearch_service_domains = Resources.*[ Type == %OPENSEARCH_SERVICE_DOMAIN_TYPE ]

# Primary Rules

rule opensearch_https_required_check when is_cfn_template(%INPUT_DOCUMENT)
  %opensearch_service_domains not empty {
    check(%opensearch_service_domains.Properties)
    <<
    [CT.OPENSEARCH.PR.16]: Require an Amazon OpenSearch Service domain to use TLSv1.2
    [FIX]: Within 'DomainEndpointOptions', set 'EnforceHTTPS' to 'true' and set
    'TLSSecurityPolicy' to 'Policy-Min-TLS-1-2-2019-07'.
    >>
  }

rule opensearch_https_required_check when is_cfn_hook(%INPUT_DOCUMENT, %OPENSEARCH_SERVICE_DOMAIN_TYPE) {
  check(%INPUT_DOCUMENT.%OPENSEARCH_SERVICE_DOMAIN_TYPE.resourceProperties)
  <<
  [CT.OPENSEARCH.PR.16]: Require an Amazon OpenSearch Service domain to use TLSv1.2
  [FIX]: Within 'DomainEndpointOptions', set 'EnforceHTTPS' to 'true' and set
  'TLSSecurityPolicy' to 'Policy-Min-TLS-1-2-2019-07'.
  >>
}

# Parameterized Rules

rule check(opensearch_service_domain) {
  %opensearch_service_domain {
    # Scenario 2
    DomainEndpointOptions exists
    DomainEndpointOptions is_struct
    DomainEndpointOptions {
      # Scenarios 3, 4, 5 and 6
      EnforceHTTPS exists
      EnforceHTTPS == true
    }
    TLSSecurityPolicy exists
    TLSSecurityPolicy in %ALLOWED_TLS_POLICIES
  }
}

# Utility Rules

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.OPENSEARCH.PR.16 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.
PASS Example - Use this template to verify a compliant resource creation.

Resources:
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
Properties:
  EngineVersion: OpenSearch_1.3
  ClusterConfig:
    InstanceCount: '1'
    InstanceType: t3.small.search
  EBSOptions:
    EBSEnabled: true
    Iops: '3000'
    VolumeSize: '10'
    VolumeType: gp3
  AccessPolicies:
    Version: '2012-10-17'
    Statement:
    - Effect: Deny
      Principal:
        AWS: '*'
      Action: es:*
      Resource: '*'
  DomainEndpointOptions:
    EnforceHTTPS: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
OpenSearchServiceDomain:
  Type: AWS::OpenSearchService::Domain
Properties:
  EngineVersion: OpenSearch_1.3
  ClusterConfig:
    InstanceCount: '1'
    InstanceType: t3.small.search
  EBSOptions:
    EBSEnabled: true
    Iops: '3000'
    VolumeSize: '10'
    VolumeType: gp3
  AccessPolicies:
    Version: '2012-10-17'
    Statement:
    - Effect: Deny
      Principal:
        AWS: '*'
      Action: es:*
      Resource: '*'
  DomainEndpointOptions:
    EnforceHTTPS: true

Amazon Relational Database Service (Amazon RDS) controls

Topics
• [CT.RDS.PR.1] Require that an Amazon RDS database instance is configured with multiple Availability Zones (p. 1070)

• [CT.RDS.PR.2] Require an Amazon RDS database instance or cluster to have enhanced monitoring configured (p. 1075)

• [CT.RDS.PR.3] Require an Amazon RDS cluster to have deletion protection configured (p. 1081)

• [CT.RDS.PR.4] Require an Amazon RDS database cluster to have AWS IAM database authentication configured (p. 1086)

• [CT.RDS.PR.5] Require an Amazon RDS database instance to have minor version upgrades configured (p. 1092)

• [CT.RDS.PR.6] Require an Amazon RDS database cluster to have backtracking configured (p. 1097)

• [CT.RDS.PR.7] Require Amazon RDS database instances to have IAM authentication configured (p. 1103)

• [CT.RDS.PR.8] Require an Amazon RDS database instance to have automatic backups configured (p. 1107)

• [CT.RDS.PR.9] Require an Amazon RDS database cluster to copy tags to snapshots (p. 1112)

• [CT.RDS.PR.10] Require an Amazon RDS database instance to copy tags to snapshots (p. 1118)

• [CT.RDS.PR.11] Require an Amazon RDS database instance to have a VPC configuration (p. 1122)

• [CT.RDS.PR.12] Require an Amazon RDS event subscription to have critical cluster events configured (p. 1128)

• [CT.RDS.PR.13] Require any Amazon RDS instance to have deletion protection configured (p. 1133)

• [CT.RDS.PR.14] Require an Amazon RDS database instance to have logging configured (p. 1138)

• [CT.RDS.PR.15] Require that an Amazon RDS instance does not create DB security groups (p. 1145)

• [CT.RDS.PR.16] Require an Amazon RDS database cluster to have encryption at rest configured (p. 1150)

• [CT.RDS.PR.17] Require an Amazon RDS event notification subscription to have critical database instance events configured (p. 1156)

• [CT.RDS.PR.18] Require an Amazon RDS event notification subscription to have critical database parameter group events configured (p. 1161)

• [CT.RDS.PR.19] Require an Amazon RDS event notifications subscription to have critical database security group events configured (p. 1165)

• [CT.RDS.PR.20] Require an Amazon RDS database instance not to use a database engine default port (p. 1170)

• [CT.RDS.PR.21] Require an Amazon RDS DB cluster to have a unique administrator username (p. 1176)

• [CT.RDS.PR.22] Require an Amazon RDS database instance to have a unique administrator username (p. 1181)

• [CT.RDS.PR.23] Require an Amazon RDS database instance to not be publicly accessible (p. 1186)

• [CT.RDS.PR.24] Require an Amazon RDS database instance to have encryption at rest configured (p. 1190)

• [CT.RDS.PR.25] Require an Amazon RDS database cluster to have logging configured (p. 1194)

[CT.RDS.PR.1] Require that an Amazon RDS database instance is configured with multiple Availability Zones

This control checks whether high availability is configured for your Amazon Relational Database Service (RDS) database instances.

• Control objective: Improve availability
• **Implementation:** AWS CloudFormation Guard Rule
• **Control behavior:** Proactive
• **Resource types:** AWS::RDS::DBInstance
• **AWS CloudFormation guard rule:** [CT.RDS.PR.1 rule specification (p. 1072)](#)

**Details and examples**

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.1 rule specification (p. 1072)](#)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.1 example templates (p. 1074)](#)

**Explanation**

Amazon RDS database (DB) instances should be configured for multiple Availability Zones (AZs). This configuration increases the availability of the stored data. Deployment into multiple Availability Zones allows for automated failover, in case an Availability Zone has an outage, and during regular RDS maintenance.

**Usage considerations**

• This control applies only to Amazon RDS DB engine types mariadb, mysql, oracle-ee, oracle-ee-cdb, oracle-se2, oracle-se2-cdb, postgres, sqlserver-ee, sqlserver-se, sqlserver-ex and sqlserver-web.

**Remediation for rule failure**

Set MultiAZ to true.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Instance - Example**

Amazon RDS database instance configured with multiple Availability Zones. The example is shown in JSON and in YAML.

**JSON example**

```
{
    "DBInstance": {
        "Type": "AWS::RDS::DBInstance",
        "Properties": {
            "Engine": "postgres",
            "EngineVersion": 14.2,
            "DBInstanceClass": "db.m5.large",
            "StorageType": "gp2",
            "AllocatedStorage": 5,
            "MasterUsername": {
                "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}",
            },
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}",
            },
            "MultiAZ": true
        },
        "DeletionPolicy": "Delete"
    }
}
```
YAML example

```yaml
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    MultiAZ: true
    DeletionPolicy: Delete
```

CT.RDS.PR.1 rule specification

```
# ####################################################################
##          Rule Specification          ##
# ####################################################################
#
# Rule Identifier:
#   rds_instance_multi_az_support_check
#
# Description:
#   This control checks whether high availability is configured for your Amazon Relational
#   Database Service (RDS) database instances.
#
# Reports on:
#   AWS::RDS::DBInstance
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document does not contain any RDS DB instance resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document contains an RDS DB instance resource
#     And: 'Engine' is not one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb',
#          'oracle-se2',
#          'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#          'sqlserver-web'
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
```
And: The input document contains an RDS DB instance resource
And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2',
    'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
    'sqlserver-web'
And: 'MultiAZ' has not been specified
Then: FAIL

Scenario: 4
Given: The input document contains an RDS DB instance resource
And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2',
    'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
    'sqlserver-web'
And: 'MultiAZ' has been specified
And: 'MultiAZ' has been set to bool(false)
Then: FAIL

Scenario: 5
Given: The input document contains an RDS DB instance resource
And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2',
    'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
    'sqlserver-web'
And: 'MultiAZ' has been specified
And: 'MultiAZ' has been set to bool(true)
Then: PASS

# Constants

let RDS_DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let INPUT_DOCUMENT = this
let SUPPORTED_RDS_INSTANCE_ENGINES = [
    "mariadb", "mysql", "oracle-ee", "oracle-ee-cdb", "oracle-se2",
    "oracle-se2-cdb", "postgres", "sqlserver-ee", "sqlserver-se",
    "sqlserver-ex", "sqlserver-web"
]

# Assignments

let rds_db_instances = Resources.*[ Type == %RDS_DB_INSTANCE_TYPE ]

# Primary Rules

rule rds_instance_multi_az_support_check when is_cfn_template(%INPUT_DOCUMENT) {
    %rds_db_instances not empty {
        check(%rds_db_instances.Properties) {
            [CT.RDS.PR.1]: Require that an Amazon RDS database instance is configured with
            multiple Availability Zones
            [FIX]: Set 'MultiAZ' to 'true'.
        }
    }
}

rule rds_instance_multi_az_support_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_INSTANCE_TYPE) {
    check(%INPUT_DOCUMENT,%RDS_DB_INSTANCE_TYPE.resourceProperties) {
        [CT.RDS.PR.1]: Require that an Amazon RDS database instance is configured with
        multiple Availability Zones
        [FIX]: Set 'MultiAZ' to 'true'.
    }
}
# Parameterized Rules

```java
rule check(rds_db_instance) {
    %rds_db_instance [filter_engine(this)] {
        # Scenario 3
        MultiAZ exists
        # Scenario 4 and 5
        MultiAZ == true
    }
}
rule filter_engine(db_properties) {
    %db_properties {
        # Scenario 2
        Engine exists
        Engine is_string
        Engine in %SUPPORTED_RDS_INSTANCE_ENGINES
    }
}
```

## Utility Rules

```java
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

### CT.RDS.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

**Resources:**

<table>
<thead>
<tr>
<th>DBInstanceSecret:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: AWS::SecretsManager::Secret</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>Description: RDS instance secret</td>
</tr>
<tr>
<td>GenerateSecretString:</td>
</tr>
<tr>
<td>SecretStringTemplate: <code>{&quot;username&quot;: &quot;testUser&quot;}</code></td>
</tr>
<tr>
<td>GenerateStringKey: password</td>
</tr>
<tr>
<td>PasswordLength: 22</td>
</tr>
<tr>
<td>ExcludeCharacters: &quot;@/&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DBInstance:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: AWS::RDS::DBInstance</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>Engine: postgres</td>
</tr>
<tr>
<td>EngineVersion: 14.2</td>
</tr>
<tr>
<td>DBInstanceClass: db.m5.large</td>
</tr>
<tr>
<td>StorageType: gp2</td>
</tr>
</tbody>
</table>
[CT.RDS.PR.2] Require an Amazon RDS database instance or cluster to have enhanced monitoring configured

This control checks whether enhanced monitoring is activated for Amazon Relational Database Service (RDS) instances.

- **Control objective**: Establish logging and monitoring
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::RDS::DBInstance
- **AWS CloudFormation guard rule**: [CT.RDS.PR.2 rule specification](p. 1077)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.2 rule specification](p. 1077)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.2 example templates](p. 1080)
Explanation

In Amazon RDS, enhanced monitoring facilitates a more rapid response to performance changes in underlying infrastructure. These performance changes could result in a lack of availability of the data. Enhanced monitoring provides real-time metrics of the operating system on which your RDS DB instance runs. An agent, installed on the instance, can obtain metrics more accurately than is possible from the hypervisor layer.

Enhanced monitoring metrics are useful when you want to see how different processes or threads on a database (DB) instance use the CPU.

Usage considerations

- This control applies only to Amazon RDS DB engine types aurora, aurora-mysql, aurora-postgresql, mariadb, mysql, oracle-ee, oracle-ee-cdb, oracle-se2, oracle-se2-cdb, postgres, sqlserver-ee, sqlserver-se, sqlserver-ex and sqlserver-web

Remediation for rule failure

Set MonitoringInterval to a supported value (1, 5, 10, 15, 30, 60), and set MonitoringRoleArn to the ARN of an AWS IAM role.

The examples that follow show how to implement this remediation.

Amazon RDS DB Instance - Example

Amazon RDS DB instance configured with enhanced monitoring. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "DBInstance": {
        "Type": "AWS::RDS::DBInstance",
        "Properties": {
            "Engine": "postgres",
            "EngineVersion": 14.2,
            "DBInstanceClass": "db.m5.large",
            "StorageType": "gp2",
            "AllocatedStorage": 5,
            "MasterUsername": {
                "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}"
            },
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
            },
            "MonitoringInterval": 30,
            "MonitoringRoleArn": {
                "Fn::GetAtt": [
                    "MonitoringIAMRole",
                    "Arn"
                ]
            }
        },
        "DeletionPolicy": "Delete"
    }
}
```

**YAML example**

```yaml
...
```
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{resolve:secretsmanager:${DBInstanceSecret}::username}''
    MasterUserPassword: !Sub '{resolve:secretsmanager:${DBInstanceSecret}::password}''
    MonitoringInterval: 30
    MonitoringRoleArn: !GetAtt 'MonitoringIAMRole.Arn'
    DeletionPolicy: Delete

CT.RDS.PR.2 rule specification

```bash
# # Rule Specification
# # Rule Identifier:
# #  rds_instance_enhanced_monitoring_enabled_check
# # Description:
# #  This control checks whether enhanced monitoring is activated for Amazon Relational
# #  Database Service (RDS) instances.
# # Reports on:
# #  AWS::RDS::DBInstance
# # Evaluates:
# #  AWS CloudFormation, AWS CloudFormation hook
# # Rule Parameters:
# #  None
# # Scenarios:
# #  Scenario: 1
# #    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# #      And: The input document does not contain any RDS DB instance resources
# #      Then: SKIP
# #  Scenario: 2
# #    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# #      And: The input document contains an RDS DB instance resource
# #      And: 'Engine' is not one of 'aurora', 'aurora-mysql', 'aurora-postgresql',
#             'mariadb', 'mysql',
#             'oracle-ee', 'oracle-se2', 'oracle-se1', 'oracle-se', 'postgres', 'sqlserver-
#             ee',
#             'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#      Then: SKIP
# #  Scenario: 3
# #    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# #      And: The input document contains an RDS DB instance resource
# #      And: 'Engine' is one of 'aurora', 'aurora-mysql', 'aurora-postgresql', 'mariadb',
#             'mysql',
#             'oracle-ee', 'oracle-se2', 'oracle-se1', 'oracle-se', 'postgres', 'sqlserver-
#             ee',
```
# Constants

```javascript
let RDS_DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let INPUT_DOCUMENT = this
let SUPPORTED_RDS_INSTANCE_ENGINES = [
  "aurora", "aurora-mysql", "aurora-postgresql", "mariadb",
  "mysql", "oracle-ee", "oracle-se2", "oracle-se1", "oracle-se", "postgres", "sqlserver-ee",
  "sqlserver-se", "sqlserver-ex", "sqlserver-web"
]```

# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an RDS DB instance resource
#           And: 'Engine' is one of 'aurora', 'aurora-mysql', 'aurora-postgresql', 'mariadb',
#                  'mysql', 'oracle-ee', 'oracle-se2', 'oracle-se1', 'oracle-se', 'postgres', 'sqlserver-ee',
#                  'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#     And: 'MonitoringInterval' has been specified
#           And: 'MonitoringInterval' has not been set to a value from the list 1, 5, 10, 15, 30, 60
#     Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an RDS DB instance resource
#           And: 'Engine' is one of 'aurora', 'aurora-mysql', 'aurora-postgresql', 'mariadb',
#                  'mysql', 'oracle-ee', 'oracle-se2', 'oracle-se1', 'oracle-se', 'postgres', 'sqlserver-ee',
#                  'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#     And: 'MonitoringInterval' has been specified
#           And: 'MonitoringInterval' has been set to a value from the list 1, 5, 10, 15, 30, 60
#           And: 'MonitoringRoleArn' has not been specified or specified as an empty string
#     Then: FAIL
# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an RDS DB instance resource
#           And: 'Engine' is one of 'aurora', 'aurora-mysql', 'aurora-postgresql', 'mariadb',
#                  'mysql', 'oracle-ee', 'oracle-se2', 'oracle-se1', 'oracle-se', 'postgres', 'sqlserver-ee',
#                  'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#     And: 'MonitoringInterval' has been specified
#           And: 'MonitoringInterval' has been set to a value from the list 1, 5, 10, 15, 30, 60
#           And: 'MonitoringRoleArn' has been specified with a non-empty string or valid local reference
#     Then: PASS

AWS Control Tower User Guide
Proactive controls
"postgres", "sqlserver-ee", "sqlserver-se", "sqlserver-ex", "sqlserver-web"
]

let ALLOWED_EM_VALUES = [1, 5, 10, 15, 30, 60]

# Assignments

let rds_db_instances = Resources.*[ Type == %RDS_DB_INSTANCE_TYPE ]

# Primary Rules

rule rds_instance_enhanced_monitoring_enabled_check when is_cfn_template(%INPUT_DOCUMENT) {
    %rds_db_instances not empty {
        check(%rds_db_instances.Properties)
        << [CT.RDS.PR.2]: Require an Amazon RDS database instance or cluster to have enhanced monitoring configured.
        [FIX]: Set 'MonitoringInterval' to a supported value (1, 5, 10, 15, 30, 60), and set 'MonitoringRoleArn' to the ARN of an AWS IAM role.
    }
}

rule rds_instance_enhanced_monitoring_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_INSTANCE_TYPE) {
    check(%INPUT_DOCUMENT.%RDS_DB_INSTANCE_TYPE.resourceProperties)
    << [CT.RDS.PR.2]: Require an Amazon RDS database instance or cluster to have enhanced monitoring configured.
    [FIX]: Set 'MonitoringInterval' to a supported value (1, 5, 10, 15, 30, 60), and set 'MonitoringRoleArn' to the ARN of an AWS IAM role.
}

# Parameterized Rules

rule check(rds_db_instance) {
    %rds_db_instance [filter_engine(this)] {
        # Scenario: 3, 4, 5, 6 and 7
        MonitoringInterval exists
        MonitoringInterval in %ALLOWED_EM_VALUES
        # Scenario: 6 and 7
        MonitoringRoleArn exists
        check_for_valid_monitor_role_arn(MonitoringRoleArn)
    }
}

rule filter_engine(db_properties) {
    %db_properties {
        # Scenario: 2
        Engine exists
        Engine is_string
        Engine in %SUPPORTED_RDS_INSTANCE_ENGINES
    }
}

rule check_for_valid_monitor_role_arn(iam_role_arn) {
    %iam_role_arn {
        check_is_string_and_not_empty(this) or
        check_local_references(%INPUT_DOCUMENT, this, "AWS::IAM::Role")
    }
}

#
CT.RDS.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  MonitoringIAMRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: "Allow"
            Principal:
              Service:
                - "monitoring.rds.amazonaws.com"
            Action:
              - "sts:AssumeRole"
            Path: "/"
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/service-role/AmazonRDSEnhancedMonitoringRole
```
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

**[CT.RDS.PR.3]** Require an Amazon RDS cluster to have deletion protection configured

This control checks whether your Amazon Relational Database Service (Amazon RDS) cluster has deletion protection activated.
• **Control objective:** Improve availability
• **Implementation:** AWS CloudFormation Guard Rule
• **Control behavior:** Proactive
• **Resource types:** AWS::RDS::DBCluster
• **AWS CloudFormation guard rule:** CT.RDS.PR.3 rule specification (p. 1083)

Details and examples

• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.RDS.PR.3 rule specification (p. 1083)
• For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.RDS.PR.3 example templates (p. 1084)

Explanation

Enabling cluster deletion protection is an additional layer of protection against accidental database deletion or deletion by an unauthorized entity.

When deletion protection is enabled, an Amazon RDS cluster cannot be deleted. Before a deletion request can succeed, deletion protection must be deactivated.

**Remediation for rule failure**

Set the value of the DeletionProtection parameter to true.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Cluster - Example**

Amazon RDS DB cluster with deletion protection enabled. The example is shown in JSON and in YAML.

**JSON example**

```
{
   "RDSDBCluster": {
      "Type": "AWS::RDS::DBCluster",
      "Properties": {
         "Engine": "aurora",
         "MasterUsername": {
            "Fn::Sub": "{{resolve:secretsmanager:${RDSClusterSecret}::username}}"
         },
         "MasterUserPassword": {
            "Fn::Sub": "{{resolve:secretsmanager:${RDSClusterSecret}::password}}"
         },
         "DBSubnetGroupName": {
            "Ref": "TestDBSubnetGroup"
         },
         "DeletionProtection": true
      }
   }
}
```

**YAML example**

```
RDSDBCluster:
```
Type: AWS::RDS::DBCluster

Properties:
  Engine: aurora
  MasterUsername: !Sub '{{resolve:secretsmanager:${RDSClusterSecret}::username}}'
  MasterUserPassword: !Sub '{{resolve:secretsmanager:${RDSClusterSecret}::password}}'
  DBSubnetGroupName: !Ref 'TestDBSubnetGroup'
  DeletionProtection: true

CT.RDS.PR.3 rule specification

```plaintext
# ######################################################################
##       Rule Specification        
# ######################################################################
#
# Rule Identifier:
#   rds_cluster_deletion_protection_enabled_check
#
# Description:
#   Checks if an Amazon Relational Database Service (Amazon RDS) cluster has deletion protection enabled.
#
# Reports on:
#   AWS::RDS::DBCluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#      Given: The input document is an AWS CloudFormation or CloudFormation hook document
#      And: The input document does not contain any RDS DB cluster resources
#      Then: SKIP
#   Scenario: 2
#      Given: The input document is an AWS CloudFormation or CloudFormation hook document
#      And: The input document contains an RDS DB cluster resource
#      And: 'DeletionProtection' has not been specified
#      Then: FAIL
#   Scenario: 3
#      Given: The input document is an AWS CloudFormation or CloudFormation hook document
#      And: The input document contains an RDS DB cluster resource
#      And: 'DeletionProtection' has been specified
#      And: 'DeletionProtection' has been set to bool(false)
#      Then: FAIL
#   Scenario: 4
#      Given: The input document is an AWS CloudFormation or CloudFormation hook document
#      And: The input document contains an RDS DB cluster resource
#      And: 'DeletionProtection' has been specified
#      And: 'DeletionProtection' has been set to bool(true)
#      Then: PASS
#
# Constants
# let RDS_DB_CLUSTER_TYPE = "AWS::RDS::DBCluster"
let INPUT_DOCUMENT = this
#```
# Assignments

let db_clusters = Resources.*[ Type == %RDS_DB_CLUSTER_TYPE ]

# Primary Rules

# Primary Rules

rule rds_cluster_deletion_protection_enabled_check when is_cfn_template(%INPUT_DOCUMENT) {
  %db_clusters not empty {
    check(%db_clusters.Properties)
    <<<
        [CT.RDS.PR.3]: Require an Amazon RDS cluster to have deletion protection configured
        [FIX]: Set the value of the 'DeletionProtection' parameter to true.
    >>>
  }
}

rule rds_cluster_deletion_protection_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_CLUSTER_TYPE) {
  check(%INPUT_DOCUMENT.%RDS_DB_CLUSTER_TYPE.resourceProperties)
  <<<
      [CT.RDS.PR.3]: Require an Amazon RDS cluster to have deletion protection configured
      [FIX]: Set the value of the 'DeletionProtection' parameter to true.
     >>>
}

rule check(properties) {
  %properties {
    # Scenario 2
    DeletionProtection exists
    # Scenario 3 and 4
    DeletionProtection == true
  }
}

# Utility Rules

# Utility Rules

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists  or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.RDS.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16
    EnableDnsHostnames: true
    EnableDnsSupport: true
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.
CidrBlock: 10.0.0.0/25
AvailabilityZone:
  Fn::Select:
    - 0
    - Fn::GetAZs: ''
SubnetTwo:
  Type: AWS::EC2::Subnet
  Properties:
    CidrBlock: 10.0.0.128/25
    AvailabilityZone:
      Fn::Select:
        - 1
        - Fn::GetAZs: ''
    VpcId:
      Ref: VPC
DBSubnetGroup:
  Type: AWS::RDS::DBSubnetGroup
  Properties:
    DBSubnetGroupDescription: Example DB subnet group
    SubnetIds:
      - Ref: SubnetOne
      - Ref: SubnetTwo
RDSClusterSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: RDS cluster secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "exampleuser"}
      GenerateStringKey: password
      PasswordLength: 32
      ExcludeCharacters: "/@"'
RDSCluster:
  Type: AWS::RDS::DBCluster
  Properties:
    Engine: aurora-mysql
    MasterUsername:
      Fn::Sub: "{{resolve:secretsmanager:${RDSClusterSecret}::username}}"
    MasterUserPassword:
      Fn::Sub: "{{resolve:secretsmanager:${RDSClusterSecret}::password}}"
    DBSubnetGroupName:
      Ref: DBSubnetGroup
    DeletionProtection: false

**[CT.RDS.PR.4] Require an Amazon RDS database cluster to have AWS IAM database authentication configured**

This control checks whether an Amazon Relational Database Service (RDS) database (DB) cluster has AWS IAM database authentication activated.

- **Control objective:** Use strong authentication
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBCluster
- **AWS CloudFormation guard rule:** [CT.RDS.PR.4 rule specification](p. 1088)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.4 rule specification](p. 1088)
For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.RDS.PR.4 example templates (p. 1090)

**Explanation**

IAM database authentication allows for password-free authentication to database instances. The authentication uses an authentication token. Network traffic to and from the database is encrypted using SSL.

**Usage considerations**

- This control applies only to Amazon RDS DB cluster engine types aurora, aurora-mysql and aurora-postgresql.

**Remediation for rule failure**

Set `EnableIAMDatabaseAuthentication` to `true`.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Cluster - Example**

Amazon RDS DB cluster configured with AWS IAM database authentication. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "DBCluster": {
        "Type": "AWS::RDS::DBCluster",
        "Properties": {
            "Engine": "aurora-mysql",
            "MasterUsername": {
                "Fn::Sub": "{{resolve:secretsmanager:${DBClusterSecret}::username}}" 
            },
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager:${DBClusterSecret}::password}}"
            },
            "DBSubnetGroupName": {
                "Ref": "DBSubnetGroup"
            },
            "EnableIAMDatabaseAuthentication": true
        }
    }
}
```

**YAML example**

```yaml
DBCluster:
    Type: AWS::RDS::DBCluster
    Properties:
        Engine: aurora-mysql
        MasterUsername: !Sub '{{resolve:secretsmanager:${DBClusterSecret}::username}}'
        MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBClusterSecret}::password}}'
        DBSubnetGroupName: !Ref 'DBSubnetGroup'
        EnableIAMDatabaseAuthentication: true
```
CT.RDS.PR.4 rule specification

# ###################################################################
##       Rule Specification        ##
###################################################################
#
# Rule Identifier:
#   rds_cluster_iam_authentication_enabled_check
#
# Description:
#   This control checks whether an Amazon Relational Database Service (RDS) database (DB) cluster has AWS IAM database authentication activated.
#
# Reports on:
#   AWS::RDS::DBCluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook document
#     And: The input document does not contain any RDS DB cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook document
#     And: The input document contains an RDS DB cluster resource
#     And: 'Engine' provided is not one of 'aurora' or 'aurora-mysql' or 'aurora-postgresql'
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook document
#     And: The input document contains an RDS DB cluster resource
#     And: 'Engine' provided is one of 'aurora' or 'aurora-mysql' or 'aurora-postgresql'
#     And: 'EnableIAMDatabaseAuthentication' has not been provided
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook document
#     And: The input document contains an RDS DB cluster resource
#     And: 'Engine' provided is one of 'aurora' or 'aurora-mysql' or 'aurora-postgresql'
#     And: 'EnableIAMDatabaseAuthentication' has been provided
#     And: 'EnableIAMDatabaseAuthentication' has been set to a value other than bool(true)
#     Then: FAIL
#   Scenario: 5
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook document
#     And: The input document contains an RDS DB cluster resource
#     And: 'Engine' provided is one of 'aurora' or 'aurora-mysql' or 'aurora-postgresql'
#     And: 'EnableIAMDatabaseAuthentication' has been provided
#     And: 'EnableIAMDatabaseAuthentication' has been set to bool(true)
#     Then: PASS
#
# Constants
#
let RDS_DB_CLUSTERS_TYPE = "AWS::RDS::DBCluster"
let SUPPORTED_DB_CLUSTER_ENGINES = ["aurora", "aurora-mysql", "aurora-postgresql"]
let INPUT_DOCUMENT = this

# # Assignments
#
let db_clusters = Resources.*[ Type == %RDS_DB_CLUSTER_TYPE ]

#
# Primary Rules
#
rule rds_cluster_iam_authentication_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%db_clusters not empty {
  check(%db_clusters.Properties)
  %db_clusters not empty {
    [CT.RDS.PR.4]: Require an Amazon RDS database cluster to have AWS IAM database authentication configured
    [FIX]: Set 'EnableIAMDatabaseAuthentication' to 'true'.
  }!
}
rule rds_cluster_iam_authentication_enabled_check when is_cfn_hook(%INPUT_DOCUMENT,
%RDS_DB_CLUSTER_TYPE) {
  check(%INPUT_DOCUMENT.%RDS_DB_CLUSTER_TYPE.resourceProperties)
  %db_clusters not empty {
    [CT.RDS.PR.4]: Require an Amazon RDS database cluster to have AWS IAM database authentication configured
    [FIX]: Set 'EnableIAMDatabaseAuthentication' to 'true'.
  }!
}

rule check(db_cluster) {
  %db_cluster [
    # Scenario 2
    filter_engine(this)
  ] {
    # Scenario 3
    EnableIAMDatabaseAuthentication exists
    # Scenario 4 and 5
    EnableIAMDatabaseAuthentication == true
  }
}

rule filter_engine(cluster_properties) {
  %cluster_properties {
    Engine exists
    Engine in %SUPPORTED_DB_CLUSTER_ENGINES
  }
}

# # Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.RDS.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
  SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/25
      AvailabilityZone: 
        Fn::Select:
          - 0
          - Fn::GetAZs: ''
  SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.0.128/25
      AvailabilityZone: 
        Fn::Select:
          - 1
          - Fn::GetAZs: ''
      VpcId:
        Ref: VPC
  DBSubnetGroup:
    Type: AWS::RDS::DBSubnetGroup
    Properties:
      DBSubnetGroupDescription: DB subnet group for DBCluster
      SubnetIds:
        - Ref: SubnetOne
        - Ref: SubnetTwo
  DBClusterSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: RDS DB cluster secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "examplemasteruser"}'
        GenerateStringKey: password
        PasswordLength: 32
        ExcludeCharacters: "/@"'\"
  DBCluster:
    Type: AWS::RDS::DBCluster
    Properties:
      Engine: aurora-mysql
      MasterUsername:
        Fn::Sub: '${resolve:secretsmanager:${DBClusterSecret}::username}"
      MasterUserPassword:
        Fn::Sub: '${resolve:secretsmanager:${DBClusterSecret}::password}'
      DBSubnetGroupName:
        Ref: DBSubnetGroup
      EnableIAMDatabaseAuthentication: true
```
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:

VPC:
Type: AWS::EC2::VPC
Properties:
  CidrBlock: 10.0.0.0/16
  EnableDnsHostnames: true
  EnableDnsSupport: true

SubnetOne:
Type: AWS::EC2::Subnet
Properties:
  VpcId: Ref: VPC
  CidrBlock: 10.0.0.0/25
  AvailabilityZone:
    Fn::Select:
      - 0
      - Fn::GetAZs: ''

SubnetTwo:
Type: AWS::EC2::Subnet
Properties:
  CidrBlock: 10.0.0.128/25
  AvailabilityZone:
    Fn::Select:
      - 1
      - Fn::GetAZs: ''
  VpcId:
    Ref: VPC

DBSubnetGroup:
Type: AWS::RDS::DBSubnetGroup
Properties:
  DBSubnetGroupDescription: DB subnet group for DBCluster
  SubnetIds:
    - Ref: SubnetOne
    - Ref: SubnetTwo

DBClusterSecret:
Type: AWS::SecretsManager::Secret
Properties:
  Description: RDS DB cluster secret
  GenerateSecretString:
    SecretStringTemplate: '{"username": "examplemasteruser"}"
    GenerateStringKey: password
    PasswordLength: 32
    ExcludeCharacters: "/@"'\""

DBCluster:
Type: AWS::RDS::DBCluster
Properties:
  Engine: aurora-mysql
  MasterUsername:
    Fn::Sub: '{{resolve:secretsmanager:${DBClusterSecret}::username}}'
  MasterUserPassword:
    Fn::Sub: '{{resolve:secretsmanager:${DBClusterSecret}::password}}'
  DBSubnetGroupName:
    Ref: DBSubnetGroup
[CT.RDS.PR.5] Require an Amazon RDS database instance to have minor version upgrades configured

This control checks whether automatic minor version upgrades are enabled for an Amazon Relational Database Service (RDS) database instance.

- **Control objective:** Manage vulnerabilities
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBInstance
- **AWS CloudFormation guard rule:** [CT.RDS.PR.5 rule specification](p. 1094)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.5 rule specification](p. 1094)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.5 example templates](p. 1096)

**Explanation**

By activating automatic minor version upgrades, you can ensure that the latest minor version updates to the relational database management system (RDBMS) are installed. These upgrades might include security patches and bug fixes. Keeping up to date with patch installation is an important step in securing systems.

**Usage considerations**

- This control applies only to Amazon RDS DB engine types aurora, aurora-mysql, aurora-postgresql, mariadb, mysql, oracle-ee, oracle-ee-cdb, oracle-se2, oracle-se2-cdb, postgres, sqlserver-ee, sqlserver-se, sqlserver-ex and sqlserver-web.

**Remediation for rule failure**

Omit the `AutoMinorVersionUpgrade` property or set it to `true`.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Instance - Example One**

Amazon RDS DB instance configured with automatic minor version upgrades, enabled by means of AWS CloudFormation defaults. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "DBInstance": {
        "Type": "AWS::RDS::DBInstance",
        "Properties": {
            "Engine": "postgres",
            "EngineVersion": 14.2,
            "DBInstanceClass": "db.m5.large",
            "StorageType": "gp2",
            "AllocatedStorage": 5,
        }
    }
}
```
"MasterUsername": {
  "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}"
},
"MasterUserPassword": {
  "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
},
"DeletionPolicy": "Delete"
}

YAML example

DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    AutoMinorVersionUpgrade: true
  DeletionPolicy: "Delete"

The examples that follow show how to implement this remediation.

Amazon RDS DB Instance - Example Two

Amazon RDS DB instance configured with automatic minor version upgrades, enabled by means of the AutoMinorVersionUpgrade property. The example is shown in JSON and in YAML.

JSON example

{
  "DBInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "Engine": "postgres",
      "EngineVersion": "14.2",
      "DBInstanceClass": "db.m5.large",
      "StorageType": "gp2",
      "AllocatedStorage": 5,
      "MasterUsername": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}"
      },
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
      },
      "AutoMinorVersionUpgrade": true
    },
    "DeletionPolicy": "Delete"
  }
}

YAML example
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    - Engine: postgres
    - EngineVersion: 14.2
    - DBInstanceClass: db.m5.large
    - StorageType: gp2
    - AllocatedStorage: 5
    - MasterUsername: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
    - MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    - AutoMinorVersionUpgrade: true
    - DeletionPolicy: Delete

CT.RDS.PR.5 rule specification

```# ###########################
##       Rule Specification        
# ###########################
#
# Rule Identifier:
#   rds_instance_automatic_minor_version_upgrade_enabled_check
#
# Description:
#   This control checks whether automatic minor version upgrades are enabled for an Amazon
#   Relational Database Service (RDS) database instance.
#
# Reports on:
#   AWS::RDS::DBInstance
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     And: The input document does not contain any RDS DB instance resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     And: The input document contains an RDS DB instance resource
#     And: 'Engine' is not one of 'aurora', 'aurora-mysql', 'aurora-postgresql',
#         'mariadb', 'mysql',
#         'oracle-ee', 'oracle-ee-cdb', 'oracle-se2', 'oracle-se2-cdb',
#         'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web',
#         'postgres'
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     And: The input document contains an RDS DB instance resource
#     And: 'Engine' is one of 'aurora', 'aurora-mysql', 'aurora-postgresql', 'mariadb',
#         'mysql',
#         'oracle-ee', 'oracle-ee-cdb', 'oracle-se2', 'oracle-se2-cdb',
#         'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web',
#         'postgres'
```
### And: 'AutoMinorVersionUpgrade' has been specified
### And: 'AutoMinorVersionUpgrade' has been set to bool(false)
### Then: FAIL

### Scenario: 4
### Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
### And: The input document contains an RDS DB instance resource
### And: 'Engine' is one of 'aurora', 'aurora-mysql', 'aurora-postgresql', 'mariadb',
### 'mysql',
### 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2', 'oracle-se2-cdb',
### 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web',
### 'postgres'
### And: 'AutoMinorVersionUpgrade' has not been specified
### Then: PASS

### Scenario: 5
### Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
### And: The input document contains an RDS DB instance resource
### And: 'Engine' is one of 'aurora', 'aurora-mysql', 'aurora-postgresql', 'mariadb',
### 'mysql',
### 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2', 'oracle-se2-cdb',
### 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web',
### 'postgres'
### And: 'AutoMinorVersionUpgrade' has been specified
### And: 'AutoMinorVersionUpgrade' has been set to bool(true)
### Then: PASS

### Constants

```scala
let RDS_DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let INPUT_DOCUMENT = this
let SUPPORTED_RDS_INSTANCE_ENGINES = [
  "aurora", "aurora-mysql", "aurora-postgresql", "mariadb", "mysql",
  "oracle-ee", "oracle-ee-cdb", "oracle-se2", "oracle-se2-cdb",
  "sqlserver-ee", "sqlserver-se", "sqlserver-ex", "sqlserver-web",
  "postgres"
]
```

### Assignments

```scala
let rds_db_instances = Resources.*[ Type == %RDS_DB_INSTANCE_TYPE ]
```

### Primary Rules

```scala
rule rds_instance_automatic_minor_version_upgrade_enabled_check when
  is_cfn_template(%INPUTDOCUMENT)
  %rds_db_instances not empty
{
  check(%rds_db_instances.Properties)
  <<
  [CT.RDS.PR.5]: Require an Amazon RDS database instance to have minor version
  upgrades configured
  [FIX]: Omit the 'AutoMinorVersionUpgrade' property or set it to 'true'.
  >>
}
```

```scala
rule rds_instance_automatic_minor_version_upgrade_enabled_check when
  is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_INSTANCE_TYPE) {
  check(%INPUT_DOCUMENT.%RDS_DB_INSTANCE_TYPE.resourceProperties)
  <<
  [CT.RDS.PR.5]: Require an Amazon RDS database instance to have minor version
  upgrades configured
  [FIX]: Omit the 'AutoMinorVersionUpgrade' property or set it to 'true'.
  >>
}```
## Proactive controls

### Parameterized Rules

```python
# Parameterized Rules
#
rule check(rds_db_instance) {
  %rds_db_instance [ filter_engine(this) ] {
    # Scenario: 4
    AutoMinorVersionUpgrade not exists or
    # Scenario: 3 and 5
    AutoMinorVersionUpgrade == true
  }
}

rule filter_engine(db_properties) {
  %db_properties {
    # Scenario: 2
    Engine exists
    Engine is_string
    Engine in %SUPPORTED_RDS_INSTANCE_ENGINES
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists  or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

### CT.RDS.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```
Resources:
DBInstanceSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: Test RDS DB Instance secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "testUser"}'
    GenerateStringKey: password
    PasswordLength: 22
    ExcludeCharacters: '"@/\'
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
```
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:

DBInstanceSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: Test RDS DB Instance secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "testUser"}'
      GenerateStringKey: password
      PasswordLength: 22
      ExcludeCharacters: '"/@/'

DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername:
      Fn::Sub: '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword:
      Fn::Sub: '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    AutoMinorVersionUpgrade: false
    DeletionPolicy: Delete

[CT.RDS.PR.6] Require an Amazon RDS database cluster to have backtracking configured

This control checks whether an Amazon Relational Database Service (RDS) database (DB) cluster has backtracking enabled.

- **Control objective:** Improve resiliency
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBCluster
- **AWS CloudFormation guard rule:** [CT.RDS.PR.6 rule specification](p. 1099)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.6 rule specification](p. 1099)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.6 example templates](p. 1101)

Explanation
Backups help you to recover more quickly from a security incident. Backups also strengthen the resilience of your systems. Aurora backtracking reduces the time required to recover a database for a specific point in time, and the recovery does not require a database restore.

**Usage considerations**

- This control applies only to Amazon RDS DB cluster engine types *aurora* and *aurora-mysql*, and to DB cluster engine modes *provisioned* and *parallelquery*.
- This control does not apply to Amazon RDS DB clusters that support Aurora Serverless V2 database instances (for example, RDS DB clusters configured with a `ServerlessV2ScalingConfiguration` and Aurora Serverless V2 compatible `EngineVersion`).

**Remediation for rule failure**

Set `BacktrackWindow` to a number between 1 and 259200.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Cluster - Example**

Amazon RDS DB cluster configured with a backtrack window of 720 seconds (12 minutes). The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "DBCluster": {
    "Type": "AWS::RDS::DBCluster",
    "Properties": {
      "Engine": "aurora-mysql",
      "MasterUsername": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBClusterSecret}::username}}"
      },
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBClusterSecret}::password}}"
      },
      "DBSubnetGroupName": {
        "Ref": "DBSubnetGroup"
      },
      "BacktrackWindow": 720
    }
  }
}
```

**YAML example**

```yaml
DBCluster:
  Type: AWS::RDS::DBCluster
  Properties:
    Engine: aurora-mysql
    MasterUsername: !Sub "{{resolve:secretsmanager:${DBClusterSecret}::username}}"
    MasterUserPassword: !Sub "{{resolve:secretsmanager:${DBClusterSecret}::password}}"
    DBSubnetGroupName: !Ref 'DBSubnetGroup'
    BacktrackWindow: 720
```
# CT.RDS.PR.6 rule specification

# ###########################################################################
##       Rule Specification        
# ###########################################################################
#
# Rule Identifier:
#   aurora_cluster_backtracking_enabled_check
#
# Description:
#   This control checks whether an Amazon Relational Database Service (RDS) database (DB)
#   cluster has backtracking enabled.
#
# Reports on:
#   AWS::RDS::DBCluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#    Scenario: 1
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#      And: The input document does not contain any RDS DB cluster resources
#      Then: SKIP
#    Scenario: 2
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#      And: The input document contains an RDS DB cluster resource
#             And: 'Engine' provided is not one of 'aurora' or 'aurora-mysql'
#      Then: SKIP
#    Scenario: 3
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#      And: The input document contains an RDS DB cluster resource
#             And: 'Engine' provided is one of 'aurora' or 'aurora-mysql'
#             And: 'EngineMode' provided is not one of 'provisioned' or 'parallelquery'
#      Then: SKIP
#    Scenario: 4
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#      And: The input document contains an RDS DB cluster resource
#             And: 'ServerlessV2ScalingConfiguration' is provided
#             And: 'Engine' provided is 'aurora-mysql'
#             And: 'EngineVersion' provided is '8.0.mysql_aurora.3.02.0' or higher
#      Then: SKIP
#    Scenario: 5
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#      And: The input document contains an RDS DB cluster resource
#             And: 'Engine' provided is one of 'aurora' or 'aurora-mysql'
#             And: 'EngineMode' is not provided or 'EngineMode' provided is one of 'provisioned' or 'parallelquery'
#             And: 'BacktrackWindow' has not been provided
#      Then: FAIL
#    Scenario: 6
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#           document
#      And: The input document contains an RDS DB cluster resource
#             And: 'Engine' provided is one of 'aurora' or 'aurora-mysql'
And: 'EngineMode' is not provided or 'EngineMode' provided is one of 'provisioned' or 'parallelquery'
And: 'BacktrackWindow' has been provided and is set to 0
Then: FAIL

# Scenario: 7
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an RDS DB cluster resource
And: 'Engine' provided is one of 'aurora' or 'aurora-mysql'
And: 'EngineMode' is not provided or 'EngineMode' provided is one of 'provisioned' or 'parallelquery'
And: 'BacktrackWindow' has been provided and is set to a value > 0
Then: PASS

# Constants

let RDS_DB_CLUSTER_TYPE = "AWS::RDS::DBCluster"
let SUPPORTED_DB_CLUSTER_ENGINES = ["aurora", "aurora-mysql"]
let SUPPORTED_DB_CLUSTER_ENGINE_MODES = ["provisioned", "parallelquery"]
let AURORA_SERVERLESS_V2_SUPPORTED_ENGINE = ["aurora-mysql"]
let AURORA_V3_SERVERLESS_V2_NOT_SUPPORTED_PATTERN = /^8\.|0\.(mysql_aurora)\.|(8\.|9\.)$/
let AURORA_V3_SERVERLESS_V2_SUPPORTED_PATTERN = /^8\.|0\.(mysql_aurora)\./
let INPUT_DOCUMENT = this

# Assignments

let db_clusters = Resources.*[ Type == %RDS_DB_CLUSTER_TYPE ]

# Primary Rules

rule aurora_cluster_backtracking_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%db_clusters not empty {
    check(%db_clusters.Properties) %db_clusters not empty {
        [CT.RDS.PR.6]: Require an Amazon RDS database cluster to have backtracking configured
        [FIX]: Set 'BacktrackWindow' to a number between '1' and '259200'.
    }
}

rule aurora_cluster_backtracking_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_CLUSTER_TYPE) {
    check(%INPUT_DOCUMENT.%RDS_DB_CLUSTER_TYPE.resourceProperties) <<
        [CT.RDS.PR.6]: Require an Amazon RDS database cluster to have backtracking configured
        [FIX]: Set 'BacktrackWindow' to a number between '1' and '259200'.
    }

rule check(db_cluster) {
    %db_cluster [ filter_engine_enginemode_and_serverless_v2(this) ] {
        # Scenario 5
        BacktrackWindow exists
        # Scenario 6 and 7
        BacktrackWindow > 0
    }
}

rule filter_engine_enginemode_and_serverless_v2(db_cluster) {
    # Scenario 2 and 3
%db_cluster {
  Engine exists
    Engine in %SUPPORTED_DB_CLUSTER_ENGINES
    EngineMode not exists or
    EngineMode in %SUPPORTED_DB_CLUSTER_ENGINE_MODES
}

# Scenario 4
%db_cluster [
  ServerlessV2ScalingConfiguration exists
    Engine in %AURORA_SERVERLESS_V2_SUPPORTED_ENGINE
] {
  EngineVersion in %AURORA_V3_SERVERLESS_V2_NOT_SUPPORTED_PATTERN or
  EngineVersion not in %AURORA_V3_SERVERLESS_V2_SUPPORTED_PATTERN
}
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists  or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.RDS.PR.6 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
  SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: VPC
      CidrBlock: 10.0.0.0/25
      AvailabilityZone:
        Fn::Select:
          - 0
          - Fn::GetAZs: ''
  SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.0.128/25
      AvailabilityZone:
        Fn::Select:
          - 1
          - Fn::GetAZs: ''
 FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.
DBSubnetGroupDescription: Example DB subnet group
SubnetIds:
- Ref: SubnetOne
- Ref: SubnetTwo

DBClusterSecret:
Type: AWS::SecretsManager::Secret
Properties:
  Description: RDS DB cluster secret
  GenerateSecretString:
    SecretStringTemplate: '{}"usernam": "examplemasteruser"'
    GenerateStringKey: password
    PasswordLength: 32
    ExcludeCharacters: "/@"\""

DBCluster:
Type: AWS::RDS::DBCluster
Properties:
  Engine: aurora-mysql
  EngineMode: provisioned
  MasterUsername:
    Fn::Sub: '{{resolve:secretsmanager:${DBClusterSecret}::username}}'
  MasterUserPassword:
    Fn::Sub: '{{resolve:secretsmanager:${DBClusterSecret}::password}}'
  DBSubnetGroupName:
    Ref: DBSubnetGroup

[CT.RDS.PR.7] Require Amazon RDS database instances to have IAM authentication configured

This control checks whether an Amazon RDS database (DB) instance has AWS Identity and Access Management (IAM) database authentication activated.

- **Control objective:** Use strong authentication
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBInstance
- **AWS CloudFormation guard rule:** CT.RDS.PR.7 rule specification (p. 1104)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.RDS.PR.7 rule specification (p. 1104)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.RDS.PR.7 example templates (p. 1106)

Explanation

IAM database authentication allows authentication to database instances with an authentication token instead of a password. Network traffic to and from the database is encrypted with SSL.

**Usage considerations**

- This control applies only to Amazon RDS DB engine types mariadb, mysql and postgres.

Remediation for rule failure

Set EnableIAMDatabaseAuthentication to true.
The examples that follow show how to implement this remediation.

**Amazon RDS DB Instance - Example**

Amazon RDS DB instance configured with IAM database authentication. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "DBInstance": {
        "Type": "AWS::RDS::DBInstance",
        "Properties": {
            "Engine": "postgres",
            "EngineVersion": "14.2",
            "DBInstanceClass": "db.m5.large",
            "StorageType": "gp2",
            "AllocatedStorage": 5,
            "MasterUsername": {
                "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}"
            },
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
            },
            "EnableIAMDatabaseAuthentication": true
        },
        "DeletionPolicy": "Delete"
    }
}
```

**YAML example**

```yaml
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    EnableIAMDatabaseAuthentication: true
  DeletionPolicy: Delete
```

**CT.RDS.PR.7 rule specification**

```bash
## Rule Specification
#
# Rule Identifier:
#   rds_instance_iam_authentication_enabled_check
#
# Description:
```
This control checks whether an Amazon RDS database (DB) instance has AWS Identity and Access Management (IAM) database authentication activated.

# Reports on:
# AWS::RDS::DBInstance
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain any RDS DB instance resources
# Then: SKIP
# Scenario: 2
# Given: The input document contains an RDS DB instance resource
# And: 'Engine' is not in-scope database engines - 'mariadb', 'mysql', 'postgres'
# Then: SKIP
# Scenario: 3
# Given: The input document contains an RDS DB instance resource
# And: 'Engine' is in-scope database engines - 'mariadb', 'mysql', 'postgres'
# And: 'EnableIAMDatabaseAuthentication' has not been specified
# Then: FAIL
# Scenario: 4
# Given: The input document contains an RDS DB instance resource
# And: 'Engine' is in-scope database engines - 'mariadb', 'mysql', 'postgres'
# And: 'EnableIAMDatabaseAuthentication' has been specified
# And: 'EnableIAMDatabaseAuthentication' has been set to bool(false)
# Then: FAIL
# Scenario: 5
# Given: The input document contains an RDS DB instance resource
# And: 'Engine' is in-scope database engines - 'mariadb', 'mysql', 'postgres'
# And: 'EnableIAMDatabaseAuthentication' has been specified
# And: 'EnableIAMDatabaseAuthentication' has been set to bool(true)
# Then: PASS
#
# Constants
#
let RDS_DB_INSTANCE_TYPE = AWS::RDS::DBInstance
let INPUT_DOCUMENT = this
let SUPPORTED_RDS_INSTANCE_ENGINES = ["mariadb", "mysql", "postgres"]
#
# Assignments
#
let rds_db_instances = Resources.*[ Type == %RDS_DB_INSTANCE_TYPE ]
#
# Primary Rules
#
rule rds_instance_iam_authentication_enabled_check when is_cfn_template(%INPUT_DOCUMENT) %rds_db_instances not empty {
  check(%rds_db_instances.Properties)
CT.RDS.PR.7 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
DBInstanceSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
Description: Test RDS DB Instance secret
GenerateSecretString:
  SecretStringTemplate: '{"username": "testUser"}'
  GenerateStringKey: password
  PasswordLength: 22
  ExcludeCharacters: '"@/\'
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername:
      Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::username}'
    MasterUserPassword:
      Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::password}'
    EnableIAMDatabaseAuthentication: true
    DeletionPolicy: Delete

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  DBInstanceSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: Test RDS DB Instance secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "testUser"}'
        GenerateStringKey: password
        PasswordLength: 22
        ExcludeCharacters: '"@/\'
  DBInstance:
    Type: AWS::RDS::DBInstance
    Properties:
      Engine: postgres
      EngineVersion: 14.2
      DBInstanceClass: db.m5.large
      StorageType: gp2
      AllocatedStorage: 5
      MasterUsername:
        Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::username}'
      MasterUserPassword:
        Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::password}'
      EnableIAMDatabaseAuthentication: false
      DeletionPolicy: Delete

[CT.RDS.PR.8] Require an Amazon RDS database instance to have automatic backups configured

This control checks whether Amazon RDS database (DB) instances have automated backups enabled, and verifies that the backup retention period is greater than or equal to seven (7) days.

- **Control objective**: Improve resiliency
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
• **Resource types:** AWS::RDS::DBInstance
• **AWS CloudFormation guard rule:** [CT.RDS.PR.8 rule specification (p. 1109)](#)

### Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.8 rule specification (p. 1109)](#)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.8 example templates (p. 1111)](#)

### Explanation

Backups help you recover more quickly from a security incident, and they strengthen the resilience of your systems. Amazon RDS provides an easy way to configure daily, full-instance volume snapshots.

#### Usage considerations

- This control applies only to Amazon RDS DB engine types mariadb, mysql, oracle-ee, oracle-ee-cdb, oracle-se2, oracle-se2-cdb, postgres, sqlserver-ee, sqlserver-se, sqlserver-ex and sqlserver-web.

### Remediation for rule failure

Set BackupRetentionPeriod to an integer value between 7 and 35 days (inclusive).

The examples that follow show how to implement this remediation.

#### Amazon RDS DB Instance - Example

Amazon RDS DB instance configured with automated backups configured and a backup retention period of 14 days. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "DBInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "Engine": "postgres",
      "EngineVersion": 14.2,
      "DBInstanceClass": "db.m5.large",
      "StorageType": "gp2",
      "AllocatedStorage": 5,
      "MasterUsername": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}"
      },
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
      },
      "BackupRetentionPeriod": 14
    },
    "DeletionPolicy": "Delete"
  }
}
```

**YAML example**

```yaml
# YAML example
```
CT.RDS.PR.8 rule specification

# ###################################
##       Rule Specification        ##
#####################################
#
# Rule Identifier:
#   rds_instance_backup_enabled_check
#
# Description:
#   This control checks whether Amazon RDS database (DB) instances have automated backups enabled, and verifies that the backup retention period is greater than or equal to seven (7) days.
#
# Reports on:
#   AWS::RDS::DBInstance
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any RDS DB instance resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document contains an RDS DB instance resource
#     And: 'Engine' is not one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2', 'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#     Then: SKIP
#   Scenario: 3
#     Given: The input document contains an RDS DB instance resource
#     And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2', 'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#     And: 'BackupRetentionPeriod' has been specified
#     And: 'BackupRetentionPeriod' has been set to 0 (backup disabled)
# Given: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2',
#       'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#       'sqlserver-web'
# And: 'BackupRetentionPeriod' has not been specified
# Then: FAIL

# Given: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2',
#       'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#       'sqlserver-web'
# And: 'BackupRetentionPeriod' has been specified
# And: 'BackupRetentionPeriod' has been set to < 7
# Then: FAIL

# Given: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2',
#       'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#       'sqlserver-web'
# And: 'BackupRetentionPeriod' has been specified
# And: 'BackupRetentionPeriod' has been set to an integer >= 7
# Then: PASS

# Constants
let RDS_DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let INPUT_DOCUMENT = this
let SUPPORTED_RDS_INSTANCE_ENGINES = [
  "mariadb", "mysql", "oracle-ee", "oracle-ee-cdb", "oracle-se2",
  "oracle-se2-cdb", "postgres", "sqlserver-ee", "sqlserver-se", "sqlserver-ex",
  "sqlserver-web"
]

# Assignments
let rds_db_instances = Resources.*[ Type == %RDS_DB_INSTANCE_TYPE ]

# Primary Rules
rule rds_instance_backup_enabled_check when is_cfn_template(%INPUT_DOCUMENT) {
  check(rds_db_instances.Properties) { %rds_db_instances not empty {
    [CT.RDS.PR.8]: Require an Amazon RDS database instance to have automatic backups configured
    [FIX]: Set 'BackupRetentionPeriod' to an integer value between 7 and 35 days (inclusive).
  }
}
rule rds_instance_backup_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_INSTANCE_TYPE) {

check(%INPUT_DOCUMENT.%RDS_DB_INSTANCE_TYPE.resourceProperties)

[CT.RDS.PR.8]: Require an Amazon RDS database instance to have automatic backups configured

[FIX]: Set 'BackupRetentionPolicy' to an integer value between 7 and 35 days (inclusive).

# Parameterized Rules

rule check(rds_db_instance) {
    %rds_db_instance [ filter_db_identifier_and_engine(this) ] {
        # Scenario: 3, 4, 5 and 6
        BackupRetentionPeriod exists
        BackupRetentionPeriod >= 7
    }
}

rule filter_db_identifier_and_engine(db_properties) {
    %db_properties {
        # Scenario: 2
        Engine exists
        Engine is_string
        Engine in %SUPPORTED_RDS_INSTANCE_ENGINES
    }
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.RDS.PR.8 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
DBInstanceSecret:
    Type: AWS::SecretsManager::Secret
Properties:
    Description: Test RDS DB Instance secret
    GenerateSecretString:
        SecretStringTemplate: '{"username": "testUser"}"
        GenerateStringKey: password
        PasswordLength: 22
        ExcludeCharacters: '@/'
Type: AWS::RDS::DBInstance
Properties:
  Engine: postgres
  EngineVersion: 14.2
  DBInstanceClass: db.m5.large
  StorageType: gp2
  AllocatedStorage: 5
  MasterUsername:
    Fn::Sub: '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
  MasterUserPassword:
    Fn::Sub: '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
  BackupRetentionPeriod: 14
  DeletionPolicy: Delete

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  DBInstanceSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: Test RDS DB Instance secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "testUser"}'
        GenerateStringKey: password
        PasswordLength: 22
        ExcludeCharacters: '"@/
  DBInstance:
    Type: AWS::RDS::DBInstance
    Properties:
      Engine: postgres
      EngineVersion: 14.2
      DBInstanceClass: db.m5.large
      StorageType: gp2
      AllocatedStorage: 5
      MasterUsername:
        Fn::Sub: '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
      MasterUserPassword:
        Fn::Sub: '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
      BackupRetentionPeriod: 4
      DeletionPolicy: Delete

[CT.RDS.PR.9] Require an Amazon RDS database cluster to copy tags to snapshots

This control checks whether an Amazon RDS database (DB) cluster is configured to copy all tags to snapshots created.

- **Control objective**: Protect configurations
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::RDS::DBCluster
- **AWS CloudFormation guard rule**: CT.RDS.PR.9 rule specification (p. 1114)

Details and examples
For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.9 rule specification](p. 1114)

For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.9 example templates](p. 1116)

**Explanation**

Identification and inventory of your infrastructure assets is a crucial aspect of governance and security. With visibility into all your Amazon RDS DB clusters, you can assess their security posture and take action on potential areas of weakness. We recommend that you tag snapshots in the same way as their parent RDS database clusters. Activating this setting ensures that snapshots inherit the tags of their parent database clusters.

**Usage considerations**

- This control applies only to Amazon RDS DB cluster engine types `aurora`, `aurora-mysql`, and `aurora-postgresql`.

**Remediation for rule failure**

Set `CopyTagsToSnapshot` to `true`.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Cluster - Example**

Amazon RDS DB cluster configured to copy tags to snapshots. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "DBCluster": {
        "Type": "AWS::RDS::DBCluster",
        "Properties": {
            "Engine": "aurora-mysql",
            "MasterUsername": {
                "Fn::Sub": "{{resolve:secretsmanager:${DBClusterSecret}::username}}"
            },
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager:${DBClusterSecret}::password}}"
            },
            "DBSubnetGroupName": {
                "Ref": "DBSubnetGroup"
            },
            "CopyTagsToSnapshot": true
        }
    }
}
```

**YAML example**

```yaml
DBCluster:
  Type: AWS::RDS::DBCluster
  Properties:
    Engine: aurora-mysql
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBClusterSecret}::username}}'
    DBSubnetGroupName: !Ref "DBSubnetGroup"
    CopyTagsToSnapshot: true
```

1113
CT.RDS.PR.9 rule specification

```plaintext
#Rule Identifier:
#   rds_cluster_copy_tags_to_snapshots_enabled_check
#
#Description:
#   This control checks whether an Amazon RDS DB cluster is configured to copy all tags to
#   snapshots created.
#
#Reports on:
#   AWS::RDS::DBCluster
#
#Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
#Rule Parameters:
#   None
#
#Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document does not contain any RDS DB cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document contains an RDS DB cluster resource
#     And: 'Engine' provided is not one of 'aurora' or 'aurora-mysql' or 'aurora-postgresql'
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document contains an RDS DB cluster resource
#     And: 'Engine' provided is one of 'aurora' or 'aurora-mysql' or 'aurora-postgresql'
#     And: 'CopyTagsToSnapshot' has not been provided
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document contains an RDS DB cluster resource
#     And: 'Engine' provided is one of 'aurora' or 'aurora-mysql' or 'aurora-postgresql'
#     And: 'CopyTagsToSnapshot' has been provided
#     And: 'CopyTagsToSnapshot' has been set to a value other than bool(true)
#     Then: FAIL
#   Scenario: 5
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#     document
#     And: The input document contains an RDS DB cluster resource
#     And: 'Engine' provided is one of 'aurora' or 'aurora-mysql' or 'aurora-postgresql'
#     And: 'CopyTagsToSnapshot' has been provided
#     And: 'CopyTagsToSnapshot' has been set to bool(true)
```
# Constants

let RDS_DB_CLUSTER_TYPE = "AWS::RDS::DBCluster"
let SUPPORTED_DB_CLUSTER_ENGINES = ["aurora", "aurora-mysql", "aurora-postgresql"]
let INPUT_DOCUMENT = this

# Assignments

let db_clusters = Resources.*[ Type == %RDS_DB_CLUSTER_TYPE ]

# Primary Rules

# Scenario 2
filter_engine(this)

# Scenario 3
CopyTagsToSnapshot exists

# Scenario 4 and 5
CopyTagsToSnapshot == true

rule rds_cluster_copy_tags_to_snapshots_enabled_check when is_cfn_template(%INPUT_DOCUMENT) {  
  check(%db_clusters.Properties)  
  {  
    [CT.RDS.PR.9]: Require an Amazon RDS database cluster to copy tags to snapshots  
    [FIX]: Set 'CopyTagsToSnapshot' to 'true'.  
  }
}

rule rds_cluster_copy_tags_to_snapshots_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_CLUSTER_TYPE) {  
  check(%INPUT_DOCUMENT.%RDS_DB_CLUSTER_TYPE.resourceProperties)  
  {  
    [CT.RDS.PR.9]: Require an Amazon RDS database cluster to copy tags to snapshots  
    [FIX]: Set 'CopyTagsToSnapshot' to 'true'.  
  }
}

rule check(db_cluster) {  
  %db_cluster [  
    filter_engine(this)  
  ] {  
    filter_engine(cluster_properties) {  
      %cluster_properties {  
        Engine exists  
        Engine in %SUPPORTED_DB_CLUSTER_ENGINES  
      }
    }
  }
}

# Utility Rules

rule is_cfn_template(doc) {  
  %doc {  
    AWSTemplateFormatVersion exists or  
    Resources exists  
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {  
  %doc.%RESOURCE_TYPE.resourceProperties exists
CT.RDS.PR.9 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
  SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: Ref: VPC
      CidrBlock: 10.0.0.0/25
      AvailabilityZone: Fn::Select:
        - 0
        - Fn::GetAZs: ''
  SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.0.128/25
      AvailabilityZone: Fn::Select:
        - 1
        - Fn::GetAZs: ''
      VpcId: Ref: VPC
  DBSubnetGroup:
    Type: AWS::RDS::DBSubnetGroup
    Properties:
      DBSubnetGroupDescription: DB subnet group for DBCluster
      SubnetIds:
        - Ref: SubnetOne
        - Ref: SubnetTwo
  DBClusterSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: RDS DB cluster secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "examplemasteruser"}'
        GenerateStringKey: password
        PasswordLength: 32
        ExcludeCharacters: "/@""
  DBCluster:
    Type: AWS::RDS::DBCluster
    Properties:
      Engine: aurora-mysql
      MasterUsername:
        Fn::Sub: '{{resolve:secretsmanager:${DBClusterSecret}::username}}'
      MasterUserPassword:
        Fn::Sub: '{{resolve:secretsmanager:${DBClusterSecret}::password}}'
      DBSubnetGroupName:
        Ref: DBSubnetGroup
```
CopyTagsToSnapshot: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

<table>
<thead>
<tr>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPC:</td>
</tr>
<tr>
<td>Type: AWS::EC2::VPC</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>CidrBlock: 10.0.0.0/16</td>
</tr>
<tr>
<td>EnableDnsHostnames: true</td>
</tr>
<tr>
<td>EnableDnsSupport: true</td>
</tr>
<tr>
<td>SubnetOne:</td>
</tr>
<tr>
<td>Type: AWS::EC2::Subnet</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>VpcId:</td>
</tr>
<tr>
<td>Ref: VPC</td>
</tr>
<tr>
<td>CidrBlock: 10.0.0.0/25</td>
</tr>
<tr>
<td>AvailabilityZone:</td>
</tr>
<tr>
<td>Fn::Select:</td>
</tr>
<tr>
<td>- 0</td>
</tr>
<tr>
<td>- Fn::GetAZs: ''</td>
</tr>
<tr>
<td>SubnetTwo:</td>
</tr>
<tr>
<td>Type: AWS::EC2::Subnet</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>CidrBlock: 10.0.0.128/25</td>
</tr>
<tr>
<td>AvailabilityZone:</td>
</tr>
<tr>
<td>Fn::Select:</td>
</tr>
<tr>
<td>- 1</td>
</tr>
<tr>
<td>- Fn::GetAZs: ''</td>
</tr>
<tr>
<td>VpcId:</td>
</tr>
<tr>
<td>Ref: VPC</td>
</tr>
<tr>
<td>DBSubnetGroup:</td>
</tr>
<tr>
<td>Type: AWS::RDS::DBSubnetGroup</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>DBSubnetGroupDescription: DB subnet group for DBCluster</td>
</tr>
<tr>
<td>SubnetIds:</td>
</tr>
<tr>
<td>- Ref: SubnetOne</td>
</tr>
<tr>
<td>- Ref: SubnetTwo</td>
</tr>
<tr>
<td>DBClusterSecret:</td>
</tr>
<tr>
<td>Type: AWS::SecretsManager::Secret</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>Description: RDS DB cluster secret</td>
</tr>
<tr>
<td>GenerateSecretString:</td>
</tr>
<tr>
<td>SecretStringTemplate: '{&quot;username&quot;: &quot;examplemasteruser&quot;}'</td>
</tr>
<tr>
<td>GenerateStringKey: password</td>
</tr>
<tr>
<td>PasswordLength: 32</td>
</tr>
<tr>
<td>ExcludeCharacters: &quot;/@&quot;'&quot;</td>
</tr>
<tr>
<td>DBCluster:</td>
</tr>
<tr>
<td>Type: AWS::RDS::DBCluster</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>Engine: aurora-mysql</td>
</tr>
<tr>
<td>MasterUsername:</td>
</tr>
<tr>
<td>Fn::Sub: '{resolve:secretsmanager:${DBClusterSecret}:username}'</td>
</tr>
<tr>
<td>MasterUserPassword:</td>
</tr>
<tr>
<td>Fn::Sub: '{resolve:secretsmanager:${DBClusterSecret}:password}'</td>
</tr>
<tr>
<td>DBSubnetGroupName:</td>
</tr>
<tr>
<td>Ref: DBSubnetGroup</td>
</tr>
</tbody>
</table>
[CT.RDS.PR.10] Require an Amazon RDS database instance to copy tags to snapshots

This control checks whether Amazon RDS database (DB) instances are configured to copy all tags to snapshots created.

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBInstance
- **AWS CloudFormation guard rule:** [CT.RDS.PR.10 rule specification (p. 1119)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.10 rule specification (p. 1119)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.10 example templates (p. 1121)]

**Explanation**

Identification and inventory of your IT assets is a crucial aspect of governance and security. With visibility of all your RDS DB instances, you can assess their security posture and take action on potential areas of weakness. Snapshots should be tagged to match their parent RDS database instances. Enabling this setting ensures that snapshots inherit the tags from their parent database instances.

**Usage considerations**

- This control applies only to Amazon RDS DB engine types mariadb, mysql, oracle-ee, oracle-ee-cdb, oracle-se2, oracle-se2-cdb, postgres, sqlserver-ee, sqlserver-se, sqlserver-ex and sqlserver-web.

**Remediation for rule failure**

Set CopyTagsToSnapshot to true.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Instance - Example**

Amazon RDS DB instance configured to copy all tags to snapshots created. The example is shown in JSON and in YAML.

**JSON example**

```json
{
"DBInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
        "Engine": "postgres",
        "EngineVersion": 14.2,
        "DBInstanceClass": "db.m5.large",
        "StorageType": "gp2",
        "AllocatedStorage": 5,
        "MasterUsername": {
            ...
```
"Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}",
"MasterUserPassword": {
  "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}",
  "CopyTagsToSnapshot": true
},
"DeletionPolicy": "Delete"
}

YAML example

```yaml
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    CopyTagsToSnapshot: true
    DeletionPolicy: Delete
```

CT.RDS.PR.10 rule specification

```bash
# ###################################################################
# Rule Specification    ##
# ###################################################################
#
# Rule Identifier:    #
#  rds_instance_copy_tags_to_snapshots_enabled_check    #
# Description:    #
#  This control checks whether Amazon RDS database (DB) instances are configured to copy all tags to snapshots created.    #
# Reports on:    #
#  AWS::RDS::DBInstance    #
# Evaluates:    #
#  AWS CloudFormation, AWS CloudFormation hook    #
# Rule Parameters:    #
#  None    #
# Scenarios:    #
#  Scenario: 1    #
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document    #
#    And: The input document does not contain any RDS DB instance resources    #
#    Then: SKIP    #
#  Scenario: 2    #
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document    #
#    And: The input document contains an RDS DB instance resource
```

1119
And: 'Engine' is not one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2', 'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#      Then: SKIP

Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an RDS DB instance resource
#     And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2', 'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#     And: 'CopyTagsToSnapshot' has not been specified
#      Then: FAIL

Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an RDS DB instance resource
#     And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2', 'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#     And: 'CopyTagsToSnapshot' has been specified
#     And: 'CopyTagsToSnapshot' has been set to bool(false)
#      Then: FAIL

Scenario: 5
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an RDS DB instance resource
#     And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2', 'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#     And: 'CopyTagsToSnapshot' has been specified
#     And: 'CopyTagsToSnapshot' has been set to bool(true)
#      Then: PASS

Constants
#
let RDS_DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let INPUT_DOCUMENT = this
let SUPPORTED_RDS_INSTANCE_ENGINES = [
]

Assignments
#
let rds_db_instances = Resources.*[ Type == %RDS_DB_INSTANCE_TYPE ]

Primary Rules
#
rule rds_instance_copy_tags_to_snapshots_enabled_check when
  is_cfn_template(%INPUT_DOCUMENT)
  %rds_db_instances not empty { check(%rds_db_instances.Properties)
    [CT.RDS.PR.10]: Require an Amazon RDS database instance to copy tags to snapshots
    [FIX]: Set 'CopyTagsToSnapshot' to 'true'.
  }
rule rds_instance_copy_tags_to_snapshots_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_INSTANCE_TYPE) {
    check(%INPUT_DOCUMENT.%RDS_DB_INSTANCE_TYPE.resourceProperties)
    <<
    [CT.RDS.PR.10]: Require an Amazon RDS database instance to copy tags to snapshots
    [FIX]: Set ‘CopyTagsToSnapshot’ to ‘true’.
    >>
}

# Parameterized Rules
#
rule check(rds_db_instance) {
    %rds_db_instance [filter_engine(this)] {
        # Scenario: 3
        CopyTagsToSnapshot exists
        # Scenario: 4 and 5
        CopyTagsToSnapshot == true
    }
}

rule filter_engine(db_properties) {
    %db_properties {
        # Scenario: 2
        Engine exists
        Engine is_string
        Engine in %SUPPORTED_RDS_INSTANCE_ENGINES
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

**CT.RDS.PR.10 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

**Resources:**
- **DBInstanceSecret:**
  - Type: AWS::SecretsManager::Secret
  - Properties:
    - Description: Test RDS DB Instance secret
    - GenerateSecretString:
      - SecretStringTemplate: '{"username": "testUser"}'
      - GenerateStringKey: password
      - PasswordLength: 22
      - ExcludeCharacters: '@\'
[CT.RDS.PR.11] Require an Amazon RDS database instance to have a VPC configuration

This control checks whether an Amazon RDS database (DB) instance is deployed in a VPC (that is, with an EC2-VPC instance).

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBInstance
- **AWS CloudFormation guard rule:** CT.RDS.PR.11 rule specification (p. 1124)

Details and examples
- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.11 rule specification (p. 1124)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.11 example templates (p. 1127)]

**Explanation**

Amazon Virtual Private Cloud (Amazon VPC) provides a number of network controls to create secure access to RDS resources. These controls include VPC endpoints, network ACLs, and security groups. To take advantage of these controls, create your Amazon RDS instances as EC2 VPC instances.

**Usage considerations**

- This control applies only to Amazon RDS DB engine types mariadb, mysql, oracle-ee, oracle-ee-cdb, oracle-se2, oracle-se2-cdb, postgres, sqlserver-ee, sqlserver-se, sqlserver-ex and sqlserver-web.

**Remediation for rule failure**

Set a DBSubnetGroupName.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Instance - Example**

Amazon RDS DB instance configured to deploy in an Amazon VPC with an RDS DB subnet group. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "DBInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "Engine": "postgres",
      "EngineVersion": 14.2,
      "DBInstanceClass": "db.m5.large",
      "StorageType": "gp2",
      "AllocatedStorage": 5,
      "MasterUsername": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}"
      },
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
      },
      "DBSubnetGroupName": {
        "Ref": "DBSubnetGroup"
      }
    },
    "DeletionPolicy": "Delete"
  }
}
```

**YAML example**

```yaml
DBInstance:
  Type: AWS::RDS::DBInstance
```
AWS Control Tower User Guide
Proactive controls

Properties:
  Engine: postgres
  EngineVersion: 14.2
  DBInstanceClass: db.m5.large
  StorageType: gp2
  AllocatedStorage: 5
  MasterUsername: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
  MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
  DBSubnetGroupName: !Ref 'DBSubnetGroup'
  DeletionPolicy: Delete

CT.RDS.PR.11 rule specification

# #######################################################################
##       Rule Specification        ##
#######################################################################
#
# Rule Identifier:
#   rds_instance_deployed_in_vpc_check
#
# Description:
#   This control checks whether an Amazon RDS database (DB) instance is deployed in a VPC
#   (that is, an EC2 VPC instance).
#
# Reports on:
#   AWS::RDS::DBInstance
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any RDS DB instance resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an RDS DB instance resource
#     And: 'Engine' is not one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb',
#          'oracle-se2',
#          'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#          'sqlserver-web'
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an RDS DB instance resource
#     And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-
#          se2',
#          'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#          'sqlserver-web'
#     And: 'DBSubnetGroupName' has not been specified
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# Proactive controls

And: The input document contains an RDS DB instance resource
And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2',
    'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
    'sqlserver-web'
And: 'DBSubnetGroupName' has been specified but is an empty string
or invalid local reference to a DB Subnet Group
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
And: The input document contains an RDS DB instance resource
And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-
    se2',
    'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
    'sqlserver-web'
And: 'DBSubnetGroupName' has been specified but is a non-empty string
or valid local reference to a DB Subnet Group
Then: PASS

# Constants

let RDS_DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let SUPPORTED_RDS_INSTANCE_ENGINES = [
    "mariadb", "mysql", "oracle-ee", "oracle-ee-cdb", "oracle-se2",
    "oracle-se2-cdb", "postgres", "sqlserver-ee", "sqlserver-se",
    "sqlserver-ex", "sqlserver-web"
]
let INPUT_DOCUMENT = this

# Assignments

let rds_db_instances = Resources.*[ Type == %RDS_DB_INSTANCE_TYPE ]

# Primary Rules

rule rds_instance_deployed_in_vpc_check when is_cfn_template(%INPUT_DOCUMENT)
%rds_db_instances not empty {
    check(%rds_db_instances.Properties)
    <<
    [CT.RDS.PR.11]: Require an Amazon RDS database instance to have a VPC configuration
    [FIX]: Set a 'DBSubnetGroupName'.
    >>
}

rule rds_instance_deployed_in_vpc_check when is_cfn_hook(%INPUT_DOCUMENT,
%RDS_DB_INSTANCE_TYPE) {
    check(%INPUT_DOCUMENT.%RDS_DB_INSTANCE_TYPE.resourceProperties)
    <<
    [CT.RDS.PR.11]: Require an Amazon RDS database instance to have a VPC configuration
    [FIX]: Set a 'DBSubnetGroupName'.
    >>
}

# Parameterized Rules

rule check(rds_db_instance) {
    %rds_db_instance [filter_engine(this)] {
    # Scenario: 3
    DBSubnetGroupName exists
    # Scenario: 4 and 5
    }
check_db_subnet_group(DBSubnetGroupName)
}
}
rule filter_engine(db_properties) {
  %db_properties {
    # Scenario: 2
    Engine exists
    Engine is_string
    Engine in %SUPPORTED_RDS_INSTANCE_ENGINES
  }
}
rule check_db_subnet_group(db_subnet_group) {
  %db_subnet_group {
    check_is_string_and_not_empty(this) or
    check_local_references(%INPUT_DOCUMENT, this, "AWS::RDS::DBSubnetGroup")
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
rule check_is_string_and_not_empty(value) {
  %value {
    this is_string
    this !/~A\s*z/
  }
}
rule check_local_references(doc, reference_properties, referenced_resource_type) {
  %reference_properties {
    'Fn::GetAtt' {
      query_for_resource(%doc, this[0], %referenced_resource_type)
      <<Local Stack reference was invalid>>
    } or Ref {
      query_for_resource(%doc, this, %referenced_resource_type)
      <<Local Stack reference was invalid>>
    }
  }
}
rule query_for_resource(doc, resource_key, referenced_resource_type) {
  let referenced_resource = %doc.Resources[ keys == %resource_key ]
  %referenced_resource not empty
  %referenced_resource {
    Type == %referenced_resource_type
  }
}
CT.RDS.PR.11 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
  SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.96.0/19
      AvailabilityZone:Fn::Select:
        - '0'
        - Fn::GetAZs: {Ref: 'AWS::Region'}
      VpcId:
        Ref: VPC
  SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.128.0/19
      AvailabilityZone:
        Fn::Select:
          - '1'
        - Fn::GetAZs: {Ref: 'AWS::Region'}
      VpcId:
        Ref: VPC
  DBSubnetGroup:
    Type: AWS::RDS::DBSubnetGroup
    Properties:
      DBSubnetGroupDescription: Test DB subnet group
      SubnetIds:
        - Ref: SubnetOne
        - Ref: SubnetTwo
  DBInstanceSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: Test RDS DB Instance secret
      GenerateSecretString:
        SecretStringTemplate: "{"username": "testUser"}"
        GenerateStringKey: password
        PasswordLength: 22
        ExcludeCharacters: '@/\'
  DBInstance:
    Type: AWS::RDS::DBInstance
    Properties:
      Engine: postgres
      EngineVersion: 14.2
      DBInstanceClass: db.m5.large
      StorageType: gp2
      AllocatedStorage: 5
      MasterUsername:
        Fn::Sub: '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
      MasterUserPassword:
        Fn::Sub: '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
      DBSubnetGroupName:
        Ref: DBSubnetGroup
      DeletionPolicy: Delete
```
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
DBInstanceSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: Test RDS DB Instance secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "testUser"}'
      GenerateStringKey: password
      PasswordLength: 22
      ExcludeCharacters: '@/\'
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername:
      Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::username}'
    MasterUserPassword:
      Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::password}'
  DeletionPolicy: Delete

[CT.RDS.PR.12] Require an Amazon RDS event subscription to have critical cluster events configured

This control checks whether your Amazon RDS event subscriptions for RDS clusters are configured to notify on event categories of maintenance and failure.

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::EventSubscription
- **AWS CloudFormation guard rule:** [CT.RDS.PR.12 rule specification (p. 1130)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.12 rule specification (p. 1130)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.12 example templates (p. 1132)]

Explanation

Amazon RDS event notifications uses Amazon SNS to make you aware of changes in the availability or configuration of your RDS resources. These notifications allow for rapid response.
Usage considerations

- This control applies only to Amazon RDS event subscriptions for RDS clusters (SourceType of db-cluster).

Remediation for rule failure

When SourceType is set to db-cluster, set Enabled to true and ensure that EventCategories contains both maintenance and failure values.

The examples that follow show how to implement this remediation.

Amazon RDS Event Subscription - Example One

Amazon RDS event subscription for RDS clusters configured to notify on all event categories. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "RDSEventSubscription": {
      "Type": "AWS::RDS::EventSubscription",
      "Properties": {
         "SnsTopicArn": {
            "Ref": "SnsTopic"
         },
         "SourceType": "db-cluster",
         "Enabled": true
      }
   }
}
```

**YAML example**

```yaml
RDSEventSubscription:
   Type: AWS::RDS::EventSubscription
   Properties:
      SnsTopicArn: !Ref 'SnsTopic'
      SourceType: db-cluster
      Enabled: true
```

The examples that follow show how to implement this remediation.

Amazon RDS Event Subscription - Example Two

Amazon RDS event subscription for RDS clusters configured to notify on maintenance and failure event categories. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "RDSEventSubscription": {
      "Type": "AWS::RDS::EventSubscription",
      "Properties": {
         "SnsTopicArn": {
            "Ref": "SnsTopic"
         },
         "EventCategories": ["maintenance", "failure"],
         "Enabled": true
      }
   }
}
```
"Properties": {
  "SnsTopicArn": {
    "Ref": "SnsTopic"
  },
  "EventCategories": [
    "maintenance",
    "failure"
  ],
  "SourceType": "db-cluster",
  "Enabled": true
}
}

YAML example

RDSEventSubscription:
  Type: AWS::RDS::EventSubscription
  Properties:
    SnsTopicArn: !Ref 'SnsTopic'
    EventCategories:
      - maintenance
      - failure
    SourceType: db-cluster
    Enabled: true

CT.RDS.PR.12 rule specification

# ###############################################################################
# Rule Specification     #
# ###############################################################################
#
# Rule Identifier:        #
#  rds_cluster_event_notifications_configured_check
#
# Description:           #
#  Checks whether an Amazon RDS event subscriptions for RDS clusters is configured to
#  notify on event categories of "maintenance" and "Failure".
#
# Reports on:            #
#  AWS::RDS::EventSubscription
#
# Evaluates:             #
#  AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:       #
#  None
#
# Scenarios:             #
#  Scenario: 1
#    Given: The input document is an AWS CloudFormation or CloudFormation hook document
#    And: The input document does not contain any Amazon RDS event subscription resources
#    Then: SKIP
#  Scenario: 2
#    Given: The input document is an AWS CloudFormation or CloudFormation hook document
#    And: The input document contains an Amazon RDS event subscription resource
#    And: 'SourceType' is provided and is not 'db-cluster'
Then: SKIP

Scenario: 3
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an Amazon RDS event subscription resource
And: 'SourceType' is 'db-cluster'
And: 'Enabled' is not provided or set to bool(false)
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an Amazon RDS event subscription resource
And: 'SourceType' is provided and is 'db-cluster'
And: 'Enabled' is provided and set to bool(true)
And: 'EventCategories' does not contain both 'maintenance' and 'failure'
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an Amazon RDS event subscription resource
And: 'SourceType' is provided and is 'db-cluster'
And: 'Enabled' is provided and set to bool(true)
And: 'EventCategories' does not exist or is an empty list
Then: PASS

Scenario: 6
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an Amazon RDS event subscription resource
And: 'SourceType' is provided and is 'db-cluster'
And: 'Enabled' is provided and set to bool(true)
And: 'EventCategories' contains both 'maintenance' and 'failure'
Then: PASS

# Constants

let RDS_EVENTSUBSCRIPTION_TYPE = "AWS::RDS::EventSubscription"
let INPUT_DOCUMENT = this
let EVENT_CATEGORIES = ["maintenance","failure"]
let EVENT_SOURCE_TYPE = "db-cluster"

# Assignments

let rds_event_subscriptions = Resources.[ Type == %RDS_EVENTSUBSCRIPTION_TYPE ]

# Primary Rules

# rule rds_cluster_event_notifications_configured_check when is_cfn_template(%INPUT_DOCUMENT)
%rds_event_subscriptions not empty {
  check(%rds_event_subscriptions.Properties) <<
  [CT.RDS.PR.12]: Require an Amazon RDS event subscription to have critical cluster events configured
  [FIX]: When 'SourceType' is set to 'db-cluster', set 'Enabled' to true and ensure that 'EventCategories' contains both 'maintenance' and 'failure' values.
  >>
}

rule rds_cluster_event_notifications_configured_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_EVENTSUBSCRIPTION_TYPE) {
  check(%INPUT_DOCUMENT.%RDS_EVENTSUBSCRIPTION_TYPE.resourceProperties) <<
  [CT.RDS.PR.12]: Require an Amazon RDS event subscription to have critical cluster events configured
  [FIX]: When 'SourceType' is set to 'db-cluster', set 'Enabled' to true and ensure that 'EventCategories' contains both 'maintenance' and 'failure' values.
  >>
}
CT.RDS.PR.12 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  SNSTopic:
    Type: AWS::SNS::Topic
    Properties: {}
  RDSEventSubscription:
    Type: AWS::RDS::EventSubscription
    Properties:
      SnsTopicArn:
        Ref: SNSTopic
      SourceType: db-cluster
      Enabled: true
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.
Resources:
SNSTopic:
  Type: AWS::SNS::Topic
  Properties: {}
RDSEventSubscription:
  Type: AWS::RDS::EventSubscription
  Properties:
    SnsTopicArn:
      Ref: SNSTopic
    EventCategories:
      - maintenance
      - deletion
    SourceType: db-cluster
    Enabled: true

[CT.RDS.PR.13] Require any Amazon RDS instance to have deletion protection configured

This control checks whether an Amazon Relational Database Service (Amazon RDS) instance has deletion protection activated.

- **Control objective**: Improve availability
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::RDS::DBInstance
- **AWS CloudFormation guard rule**: [CT.RDS.PR.13 rule specification](p. 1134)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.13 rule specification](p. 1134)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.13 example templates](p. 1137)

**Explanation**

When active, instance deletion protection provides an additional layer of protection against accidental database deletion, or deletion by an unauthorized entity.

While deletion protection is active, an RDS DB instance cannot be deleted. Before a deletion request can succeed, deletion protection must be turned off.

**Usage considerations**

- This control applies only to Amazon RDS DB engine types mariadb, mysql, oracle-ee, oracle-ee-cdb, oracle-se2, oracle-se2-cdb, postgres, sqlserver-ee, sqlserver-se, sqlserver-ex and sqlserver-web.

**Remediation for rule failure**

Set DeletionProtection to true.

The examples that follow show how to implement this remediation.
Amazon RDS DB instance - Example

Amazon RDS DB instance configured with deletion protection active. The example is shown in JSON and in YAML.

JSON example

```
{
  "DBInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "Engine": "postgres",
      "EngineVersion": 5.7,
      "DBInstanceClass": "db.m5.large",
      "StorageType": "gp2",
      "AllocatedStorage": 5,
      "MasterUsername": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}"
      },
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
      },
      "StorageEncrypted": true,
      "DeletionProtection": true
    },
    "DeletionPolicy": "Delete"
  }
}
```

YAML example

```
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 5.7
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    StorageEncrypted: true
    DeletionProtection: true
  DeletionPolicy: Delete
```

CT.RDS.PR.13 rule specification

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
## Rule Specification ##
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Rule Identifier: #
# rds_instance_deletion_protection_enabled_check  #
# Description:
# This control checks whether an Amazon Relational Database Service (Amazon RDS) instance has deletion protection activated.
```

1134
# Reports on:
# AWS::RDS::DBInstance

# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document does not contain any RDS DB instance resources
#   Then: SKIP
# Scenario: 2
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains an RDS DB instance resource
#   And: 'Engine' is not one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2',
#       'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#       'sqlserver-web'
#   Then: SKIP
# Scenario: 3
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains an RDS DB instance resource
#   And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2',
#       'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#       'sqlserver-web'
#   And: 'DeletionProtection' has not been specified
#   Then: FAIL
# Scenario: 4
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains an RDS DB instance resource
#   And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2',
#       'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#       'sqlserver-web'
#   And: 'DeletionProtection' has been specified
#   And: 'DeletionProtection' has been set to bool(false)
#   Then: FAIL
# Scenario: 5
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#   And: The input document contains an RDS DB instance resource
#   And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-se2',
#       'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#       'sqlserver-web'
#   And: 'DeletionProtection' has been specified
#   And: 'DeletionProtection' has been set to bool(true)
#   Then: PASS

# Constants

let RDS_DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let INPUT_DOCUMENT = this
let SUPPORTED_RDS_INSTANCE_ENGINES = ["mariadb", "mysql", "oracle-ee", "oracle-ee-cdb", "oracle-se2",
"oracle-se2-cdb", "postgres", "sqlserver-ee", "sqlserver-se", "sqlserver-ex", "sqlserver-web"]
"sqlserver-ex", "sqlserver-web"
]

# Assignments
let rds_db_instances = Resources.[ Type == %RDS_DB_INSTANCE_TYPE ]

# Primary Rules
rule rds_instance_deletion_protection_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
  %rds_db_instances not empty {
    check(%rds_db_instances.Properties)
    <<
    [CT.RDS.PR.13]: Require any Amazon RDS instance to have deletion protection configured
    [FIX]: Set 'DeletionProtection' to 'true'.
    >>
  }
rule rds_instance_deletion_protection_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_INSTANCE_TYPE) {
  check(%INPUT_DOCUMENT.%RDS_DB_INSTANCE_TYPE.resourceProperties)
  <<
  [CT.RDS.PR.13]: Require any Amazon RDS instance to have deletion protection configured
  [FIX]: Set 'DeletionProtection' to 'true'.
  >>
}

# Parameterized Rules
rule check(rds_db_instance) {
  %rds_db_instance [filter_engine(this)] {
  #Scenario: 3
  DeletionProtection exists
  #Scenario: 4 and 5
  DeletionProtection == true
  }
}
rule filter_engine(db_properties) {
  %db_properties {
  #Scenario: 2
  Engine exists
  Engine is_string
  Engine in %SUPPORTED_RDS_INSTANCE_ENGINES
  }
}

# Utility Rules
rule is_cfn_template(doc) {
  %doc {
  AWSTemplateFormatVersion exists or
  Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.RDS.PR.13 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
DBInstanceSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: RDS DB Instance secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "testUser"}'
      GenerateStringKey: password
      PasswordLength: 22
      ExcludeCharacters: '@/

DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername:
      Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::username}'
    MasterUserPassword:
      Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::password}'
    DeletionProtection: true
    DeletionPolicy: Delete
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
DBInstanceSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: RDS DB Instance secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "testUser"}'
      GenerateStringKey: password
      PasswordLength: 22
      ExcludeCharacters: '@/

DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername:
      Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::username}'
    MasterUserPassword:
      Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::password}'
    DeletionProtection: false
```

1137
[CT.RDS.PR.14] Require an Amazon RDS database instance to have logging configured

This rules checks whether Amazon Relational Database Service (RDS) instances have all available log types configured for export to Amazon CloudWatch Logs.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBInstance
- **AWS CloudFormation guard rule:** [CT.RDS.PR.14 rule specification (p. 1140)](CT.RDS.PR.14 rule specification (p. 1140))

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.14 rule specification (p. 1140)](CT.RDS.PR.14 rule specification (p. 1140))
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.14 example templates (p. 1144)](CT.RDS.PR.14 example templates (p. 1144))

**Explanation**

We recommend that you enable relevant logs for all Amazon RDS databases. Database logging provides detailed records of requests made to RDS. Database logs can assist with security and access audits, and they can help you diagnose availability issues.

**Usage considerations**

- This control applies only to Amazon RDS DB engine types mariadb, mysql, postgres, sqlserver-ee, sqlserver-se, sqlserver-ex, sqlserver-web, oracle-ee, oracle-se2, oracle-se1, and oracle-se.

**Remediation for rule failure**

Specify `EnableCloudwatchLogsExports` with a list of all supported log types for the Amazon RDS database instance engine.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Instance - Example One**

Amazon RDS DB instance configured with an engine type of `mysql` and all supported log types, for the `mysql` engine type. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "DBInstance": {
        "Type": "AWS::RDS::DBInstance",
        "Properties": {
            "Engine": "mysql",
```
"EngineVersion": 5.7,
"DBInstanceClass": "db.m5.large",
"StorageType": "gp2",
"AllocatedStorage": 5,
"MasterUsername": {
    "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}",
},
"MasterUserPassword": {
    "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}",
},
"StorageEncrypted": true,
"EnableCloudwatchLogsExports": [
    "error",
    "general",
    "slowquery",
    "audit"
],
"DeletionPolicy": "Delete"
}

YAML example

DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: mysql
    EngineVersion: 5.7
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    StorageEncrypted: true
    EnableCloudwatchLogsExports:
      - error
      - general
      - slowquery
      - audit
    DeletionPolicy: Delete

The examples that follow show how to implement this remediation.

Amazon RDS DB Instance - Example Two

Amazon RDS DB instance configured with an engine type of postgres and all supported log types, for the postgres engine type. The example is shown in JSON and in YAML.

JSON example

```json
[
    "DBInstance": {
        "Type": "AWS::RDS::DBInstance",
        "Properties": {
            "Engine": "postgres",
            "EngineVersion": "14.2",
```
"DBInstanceClass": "db.m5.large",
"StorageType": "gp2",
"AllocatedStorage": 5,
"MasterUsername": {
    "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}"
},
"MasterUserPassword": {
    "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
},
"EnableCloudwatchLogsExports": [
    "postgresql",
    "upgrade"
],
"DeletionPolicy": "Delete"
}

YAML example

DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    EnableCloudwatchLogsExports:
      - postgresql
      - upgrade
    DeletionPolicy: Delete

CT.RDS.PR.14 rule specification

# ####################################################################
##       Rule Specification       ##
# ####################################################################
#
# Rule Identifier:
#  rds_instance_logging_enabled_check
#
# Description:
#  This rules checks whether Amazon Relational Database Service (RDS) instances have all
#  available log types configured for export to Amazon CloudWatch Logs.
#
# Reports on:
#  AWS::RDS::DBInstance
#
# Evaluates:
#  AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#  None
#
# Scenarios:

1140
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document does not contain any RDS DB instance resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is not one of 'mariadb', 'mysql',
#   'oracle-ee', 'oracle-se2', 'oracle-ee-cdb', 'oracle-se2-cdb',
#   'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'mariadb', 'mysql',
#   'oracle-ee', 'oracle-se2', 'oracle-ee-cdb', 'oracle-se2-cdb',
#   'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
# And: 'EnableCloudwatchLogsExports' has not been specified or has been specified
#      and is an empty list
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'mariadb', 'mysql',
#   'oracle-ee', 'oracle-se2', 'oracle-ee-cdb', 'oracle-se2-cdb',
#   'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
# And: 'EnableCloudwatchLogsExports' has been specified and is a non-empty list
# And: One or more log types in 'EnableCloudwatchLogsExports' are not supported by
#      the specified 'Engine'
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'mariadb', 'mysql',
#   'oracle-ee', 'oracle-se2', 'oracle-ee-cdb', 'oracle-se2-cdb',
#   'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
# And: 'EnableCloudwatchLogsExports' has been specified and is a non-empty list
# And: 'EnableCloudwatchLogsExports' does not contain all log types supported by the
#      specified 'Engine'
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'mariadb', 'mysql'
# And: 'EnableCloudwatchLogsExports' has been specified
# And: 'EnableCloudwatchLogsExports' value is a non-empty and all supported log types
#      are enabled - 'audit', 'error', 'general', 'slowquery'
# Then: PASS
# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is 'postgres'
# And: 'EnableCloudwatchLogsExports' has been specified
# And: 'EnableCloudwatchLogsExports' value is a non-empty and all supported log types
#      are enabled - 'postgresql', 'upgrade'
# Then: PASS
# Scenario: 8
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# If: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#      'sqlserver-web'
# And: 'EnableCloudwatchLogsExports' has been specified
# And: 'EnableCloudwatchLogsExports' value is a non-empty and all supported log types
#      are enabled - 'agent', 'error'
# Then: PASS

# Scenario: 9
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'oracle-ee', 'oracle-se2', 'oracle-ee-cdb', 'oracle-se2-cdb',
#      'oracle-se1-cdb', 'oracle-se1', 'oracle-se', 'oracle-ee-cdb',
# And: 'EnableCloudwatchLogsExports' has been specified
# And: 'EnableCloudwatchLogsExports' value is a non-empty and all supported log types
#      are enabled - 'alert', 'audit', 'listener', 'oemagent', 'trace'
# Then: PASS

# Constants

let RDS_DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let INPUT_DOCUMENT = this
let SUPPORTED_RDS_INSTANCE_ENGINES = [
  "mariadb", "mysql", "oracle-ee", "oracle-ee-cdb", "oracle-se2",
  "oracle-se2-cdb", "postgres", "sqlserver-ee", "sqlserver-se",
  "sqlserver-ex", "sqlserver-web"
]

let MYSQL_OR_MARIA_ENGINES_SUBTYPES = [ "mariadb", "mysql" ]
let POSTGRES_ENGINES_SUBTYPES = [ "postgres" ]
let SQLSERVER_ENGINES_SUBTYPES = [ "sqlserver-ee", "sqlserver-se", "sqlserver-ex",
  "sqlserver-web" ]
let ORACLE_ENGINES_SUBTYPES = [ "oracle-ee", "oracle-se2", "oracle-se1", "oracle-se" ]

let MYSQL_OR_MARIA_SUPPORTED_LOG_TYPES = [ "audit", "error", "general", "slowquery" ]
let POSTGRES_SUPPORTED_LOG_TYPES = [ "postgresql", "upgrade" ]
let SQLSERVER_SUPPORTED_LOG_TYPES = [ "agent", "error" ]
let ORACLE_SUPPORTED_LOG_TYPES = [ "alert", "audit", "listener", "oemagent", "trace" ]

# Assignments

let rds_db_instances = Resources.*[ Type == %RDS_DB_INSTANCE_TYPE ]

# Primary Rules

rule rds_instance_logging_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
  %rds_db_instances not empty {
    check(%rds_db_instances.Properties)
    <<
    [CT.RDS.PR.14]: Require an Amazon RDS database instance to have logging configured
    [FIX]: Specify 'EnableCloudwatchLogsExports' with a list of all supported log types
    for the Amazon RDS database instance engine.
    >>
  }

rule rds_instance_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT,
  %RDS_DB_INSTANCE_TYPE) {
  check(%INPUT_DOCUMENT.%RDS_DB_INSTANCE_TYPE.resourceProperties)
  <<
  [CT.RDS.PR.14]: Require an Amazon RDS database instance to have logging configured
  [FIX]: Specify 'EnableCloudwatchLogsExports' with a list of all supported log types
  for the Amazon RDS database instance engine.
  >>
}
# Parameterized Rules

```
rule check(rds_db_instance) {
  %rds_db_instance [filter_engine(this)] {
    # Scenario: 3
    EnableCloudwatchLogsExports exists
    check_is_list_and_not_empty(EnableCloudwatchLogsExports)
    
    # Scenario: 4 and 5
    when Engine IN %MYSQL_OR_MARIA_ENGINES_SUBTYPES {
      %MYSQL_OR_MARIA_SUPPORTED_LOG_TYPES.* IN EnableCloudwatchLogsExports[*]
      EnableCloudwatchLogsExports.* IN %MYSQL_OR_MARIA_SUPPORTED_LOG_TYPES[*]
    }
    
    # Scenario: 4 and 6
    when Engine IN %POSTGRES_ENGINES_SUBTYPES {
      %POSTGRES_SUPPORTED_LOG_TYPES.* IN EnableCloudwatchLogsExports[*]
      EnableCloudwatchLogsExports.* IN %POSTGRES_SUPPORTED_LOG_TYPES[*]
    }
    
    # Scenario: 4 and 7
    when Engine IN %SQLSERVER_ENGINES_SUBTYPES {
      %SQLSERVER_SUPPORTED_LOG_TYPES.* IN EnableCloudwatchLogsExports[*]
      EnableCloudwatchLogsExports.* IN %SQLSERVER_SUPPORTED_LOG_TYPES[*]
    }
    
    # Scenario: 4 and 8
    when Engine IN %ORACLE_ENGINES_SUBTYPES {
      %ORACLE_SUPPORTED_LOG_TYPES.* in EnableCloudwatchLogsExports[*]
      EnableCloudwatchLogsExports.* IN %ORACLE_SUPPORTED_LOG_TYPES[*]
    }
  }
}
```

```
rule filter_engine(db_properties) {
  %db_properties {
    # Scenario: 2
    Engine exists
    Engine in %SUPPORTED_RDS_INSTANCE_ENGINES
  }
}
```

```
# Utility Rules

```
rule check_is_list_and_not_empty(value) {
  %value {
    this is_list
    this not_empty
  }
}
```

```
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
```

```
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```
CT.RDS.PR.14 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:

DBInstanceSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: RDS DB instance secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "examplemasteruser"}'
      GenerateStringKey: password
      PasswordLength: 22
      ExcludeCharacters: '@/\'

DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: mysql
    EngineVersion: 5.7
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername:
      Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::username}''
    MasterUserPassword:
      Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::password}''
    StorageEncrypted: true
    EnableCloudwatchLogsExports: ["error", "general", "slowquery", "audit"
    DeletionPolicy: Delete

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:

DBInstanceSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: RDS DB instance secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "examplemasteruser"}'
      GenerateStringKey: password
      PasswordLength: 22
      ExcludeCharacters: '@/\'

DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: mysql
    EngineVersion: 5.7
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername:
      Fn::Sub: '{resolve:secretsmanager:${DBInstanceSecret}::username}''
[CT.RDS.PR.15] Require that an Amazon RDS instance does not create DB security groups

This control checks whether any Amazon Relational Database Service (RDS) database (DB) security groups are created by, or associated to, an RDS DB instance, because DB security groups are intended for the EC2-Classic platform only.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBInstance, AWS::RDS::DBSecurityGroup
- **AWS CloudFormation guard rule:** [CT.RDS.PR.15 rule specification (p. 1146)](#)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.15 rule specification (p. 1146)](#)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.15 example templates (p. 1149)](#)

**Explanation**

We recommend that all Amazon Relational Database Service (RDS) databases use Amazon VPC security groups to secure their access. Amazon DB security groups are for the EC2-Classic platform only, and they are not recommended for use.

**Remediation for rule failure**

Omit the **DBSecurityGroups** property. Instead, configure Amazon VPC security groups by means of the **VPCSecurityGroups** property.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Instance - Example**

Amazon RDS DB instance configured with an Amazon VPC security group. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "DBInstance": {
        "Type": "AWS::RDS::DBInstance",
        "Properties": {
            "Engine": "mysql",
            "EngineVersion": "5.7",
```
"DBInstanceClass": "db.t3.small",
"StorageType": "gp2",
"AllocatedStorage": 5,
"MasterUsername": {
    "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}"
},
"MasterUserPassword": {
    "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
},
"StorageEncrypted": true,
"Port": 6733,
"DBSubnetGroupName": {
    "Ref": "DBSubnetGroup"
},
"VPCSecurityGroups": [
    "Ref": "SecurityGroup"
]
}
}}

YAML example

DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: mysql
    EngineVersion: 5.7
    DBInstanceClass: db.t3.small
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    StorageEncrypted: true
    Port: 6733
    DBSubnetGroupName: !Ref 'DBSubnetGroup'
    VPCSecurityGroups:
      - !Ref 'SecurityGroup'

CT.RDS.PR.15 rule specification

# #####################################################################
## Rule Specification
# #####################################################################
#
# Rule Identifier:
#   rds_db_security_group_not.allowed_check
#
# Description:
#   This control checks whether any Amazon Relational Database Service (RDS) database (DB)
#   security groups are created by, or associated to, an RDS DB instance, because DB security
#   groups are intended for the EC2-Class platform only.
#
# Reports on:
#   AWS::RDS::DBSecurityGroup, AWS::RDS::DBInstance
#
1146
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#      And: The input document contains a DB security group resource
#      Then: FAIL
#   Scenario: 2
#      Given: The input document contains an RDS DB instance resource
#      And: 'DBSecurityGroups' has been specified on the RDS DB instance as a non empty list
#      Then: PASS
#
# Constants
#
let DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let DB_SECURITY_GROUP_TYPE = "AWS::RDS::DBSecurityGroup"
let INPUT_DOCUMENT = this
#
# Assignments
#
let db_instances = Resources.*[ Type == %DB_INSTANCE_TYPE ]
let db_security_groups = Resources.*[ Type == %DB_SECURITY_GROUP_TYPE ]
#
# Primary Rules
#
rule rds_db_security_group_not_allowed_check when is_cfn_template(this)
  %db_security_groups not empty { check_db_security_group(%db_security_groups) } <<<
  [CT.RDS.PR.15]: Require that an Amazon RDS instance does not create DB security groups
  [FIX]: Omit the 'DBSecurityGroups' property. Instead, configure Amazon VPC security groups by means of the 'VPCSecurityGroups' property.
  >>

rule rds_db_security_group_not_allowed_check when is_cfn_template(this)
  %db_instances not empty { check_db_instance(%db_instances.Properties) } <<
[CT.RDS.PR.15]: Require that an Amazon RDS instance does not create DB security groups
  [FIX]: Omit the 'DBSecurityGroups' property. Instead, configure Amazon VPC security groups by means of the 'VPCSecurityGroups' property.
  >>>
} rule rds_db_security_group_not_allowed_check when is_cfn_hook(%INPUT_DOCUMENT, %DB_SECURITY_GROUP_TYPE) {
  check_db_security_group(%INPUT_DOCUMENT.%DB_SECURITY_GROUP_TYPE)
  <<<
} rule rds_db_security_group_not_allowed_check when is_cfn_hook(%INPUT_DOCUMENT, %DB_INSTANCE_TYPE) {
  check_db_instance(%INPUT_DOCUMENT.%DB_INSTANCE_TYPE.resourceProperties)
  <<<
} rule check_db_security_group(db_security_group) {
  # Scenario 2
  %db_security_group empty
}
} rule check_db_instance(db_instance) {
  %db_instance {
    # Scenario 3 and 4
    DBSecurityGroups not exists or
    check_is_empty_list(this)
  }
}
} rule check_is_empty_list(db_instance_configuration) {
  %db_instance_configuration {
    DBSecurityGroups is_list
    DBSecurityGroups empty
  }
}
} # Utility Rules
# rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
} rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.RDS.PR.15 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```json
Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
  SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.96.0/19
      AvailabilityZone: Fn::Select:
        - '0'
        - Fn::GetAZs: {Ref: 'AWS::Region'}
      VpcId:
        Ref: VPC
  SubnetTwo:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.128.0/19
      AvailabilityZone: Fn::Select:
        - '1'
        - Fn::GetAZs: {Ref: 'AWS::Region'}
      VpcId:
        Ref: VPC
  DBSubnetGroup:
    Type: AWS::RDS::DBSubnetGroup
    Properties:
      DBSubnetGroupDescription: DB subnet group
      SubnetIds:
        - Ref: SubnetOne
        - Ref: SubnetTwo
  DBInstanceSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: RDS DB instance secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "examplemasterusername"}'
        GenerateStringKey: password
        PasswordLength: 22
        ExcludeCharacters: '@/\'
  SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Example Security Group
      VpcId:
        Ref: VPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 6733
          ToPort: 6733
          CidrIp: 10.0.0.0/16
  DBInstance:
    Type: AWS::RDS::DBInstance
    Properties:
      Engine: mysql
      DBInstanceClass: db.t2.micro
      MasterUsername: examplemasterusername
      DBName: exampledb
      EngineVersion: 5.7.23
      Allow格尔第二代：
      MasterUserPassword: 'examplepassword'
      MasterUserPassword:
        NoPasswordSet
      VpcSecurityGroupIds:
        - Ref: SecurityGroup
      AllocatedStorage: 10
      StorageType: gp2
      MultiAz: true
      PubliclyAccessible: true
      Engine natuur
      De
      
```
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```yaml
Resources:
  DBSecurityGroup:
    Type: AWS::RDS::DBSecurityGroup
    Properties:
      DBSecurityGroupIngress:
        - CIDRIP: "0.0.0.0/0"
      GroupDescription: "Ingress for Amazon EC2 security group"
```

### [CT.RDS.PR.16] Require an Amazon RDS database cluster to have encryption at rest configured

This control checks whether the storage encryption is configured on Amazon Relational Database Service (RDS) database (DB) clusters that are not being restored from an existing cluster.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBCluster
- **AWS CloudFormation guard rule:** [CT.RDS.PR.16 rule specification (p. 1151)](#)

#### Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.16 rule specification (p. 1151)](#)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.16 example templates (p. 1155)](#)

#### Explanation

We recommend that you configure your Amazon RDS DB clusters to be encrypted at rest, to give an added layer of security for your sensitive data. To encrypt your RDS DB clusters and snapshots at rest, enable the encryption option for your RDS DB clusters. Data that is encrypted at rest includes the underlying storage for DB clusters, its automated backups, read replicas, and snapshots.
Encrypted RDS DB clusters use the open standard AES-256 encryption algorithm to encrypt your data on the server that hosts the clusters. After your data is encrypted, Amazon RDS handles authentication of access and decryption of your data with a minimal impact on performance. You do not need to modify your database client applications to use encryption.

**Usage considerations**

- This control applies only to Amazon RDS DB clusters that are not being restored from an existing cluster or created as a read replica. (For example, when `SourceDBClusterIdentifier` or `ReplicationSourceIdentifier` properties have been provided.)

**Remediation for rule failure**

Set `StorageEncrypted` to `true`.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Cluster - Example**

Amazon RDS DB cluster configured with storage encryption enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "DBCluster": {
        "Type": "AWS::RDS::DBCluster",
        "Properties": {
            "Engine": "aurora-mysql",
            "MasterUsername": {
                "Fn::Sub": "{{resolve:secretsmanager:${DBClusterSecret}::username}}"
            },
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager:${DBClusterSecret}::password}}"
            },
            "StorageEncrypted": true
        }
    }
}
```

**YAML example**

```yaml
DBCluster:
  Type: AWS::RDS::DBCluster
  Properties:
    Engine: aurora-mysql
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBClusterSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBClusterSecret}::password}}'
    StorageEncrypted: true
```

**CT.RDS.PR.16 rule specification**

```
# ################################################################
```

1151
## Rule Specification

- **Rule Identifier:** rds_cluster_storage_encrypted_check
- **Description:** This control checks whether the storage encryption is configured on Amazon Relational Database Service (RDS) database (DB) clusters that are not being restored from an existing cluster.
- **Reports on:** AWS::RDS::DBCluster
- **Evaluates:** AWS CloudFormation, AWS CloudFormation hook
- **Rule Parameters:** None
- **Scenarios:**
  - **Scenario: 1**
    - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
    - And: The input document does not contain any RDS DB cluster resources
    - Then: SKIP
  - **Scenario: 2**
    - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
    - And: The input document contains an RDS DB cluster resource
    - And: 'SourceDBClusterIdentifier' or 'ReplicationSourceIdentifier' has been provided
    - Then: SKIP
  - **Scenario: 3**
    - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
    - And: The input document contains an RDS DB cluster resource
    - And: 'SourceDBClusterIdentifier' or 'ReplicationSourceIdentifier' has not been provided
    - And: 'StorageEncrypted' has not been provided
    - Then: FAIL
  - **Scenario: 4**
    - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
    - And: The input document contains an RDS DB cluster resource
    - And: 'SourceDBClusterIdentifier' or 'ReplicationSourceIdentifier' has not been provided
    - And: 'StorageEncrypted' has been provided and set to bool(false)
    - Then: FAIL
  - **Scenario: 5**
    - Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
    - And: The input document contains an RDS DB cluster resource
    - And: 'SourceDBClusterIdentifier' or 'ReplicationSourceIdentifier' has not been provided
    - And: 'StorageEncrypted' has been provided and set to bool(true)
    - Then: PASS

### Constants

```plaintext
define RDS_CLUSTER_TYPE = "AWS::RDS::DBCluster"
define INPUT_DOCUMENT = this
```

### Assignments
AWS Control Tower User Guide
Proactive controls
#
let rds_cluster = Resources.*[ Type == %RDS_CLUSTER_TYPE ]
#
# Primary Rules
#
rule rds_cluster_storage_encrypted_check when is_cfn_template(%INPUT_DOCUMENT)
%rds_cluster not empty {
check(%rds_cluster.Properties)
<<
[CT.RDS.PR.16]: Require an Amazon RDS database cluster to have encryption at rest
configured
[FIX]: Set 'StorageEncrypted' to 'true'.
>>
}
rule rds_cluster_storage_encrypted_check when is_cfn_hook(%INPUT_DOCUMENT,
%RDS_CLUSTER_TYPE) {
check(%INPUT_DOCUMENT.%RDS_CLUSTER_TYPE.resourceProperties)
<<
[CT.RDS.PR.16]: Require an Amazon RDS database cluster to have encryption at rest
configured
[FIX]: Set 'StorageEncrypted' to 'true'.
>>
}
#
# Parameterized Rules
#
rule check(rds_cluster) {
%rds_cluster [
# Scenario 2
filter_sources(this)
] {
# Scenario 3
StorageEncrypted exists

}

}

# Scenario 4 and 5
StorageEncrypted == true

rule filter_sources(rds_cluster) {
%rds_cluster {
# Scenario 2
SourceDBClusterIdentifier not exists or
filter_property_is_empty_string(SourceDBClusterIdentifier) or
filter_is_not_valid_local_reference(%INPUT_DOCUMENT, SourceDBClusterIdentifier,
"AWS::RDS::DBCluster")

}

}

ReplicationSourceIdentifier not exists or
filter_property_is_empty_string(ReplicationSourceIdentifier) or
filter_replication_source_identifier(ReplicationSourceIdentifier)

rule filter_property_is_empty_string(value) {
%value {
this is_string
this == /\A\s*\z/
}
}
rule filter_is_not_valid_local_reference(doc, reference_properties,
referenced_resource_type) {
%reference_properties {

1153


this not is_string
this is_struct
when this.'Ref' exists {
    'Ref' {
        when query_for_resource(%doc, this, %referenced_resource_type) {
            this not exists
        } this exists
    }
    when this.'Ref' not exists {
        this exists
    }
}
when this.'Fn::Join' exists {
    'Fn::Join' {
        when filter_list_contains_valid_local_reference(this[1], %doc, %referenced_resource_type) {
            this not exists
        } this exists
    }
    when this.'Fn::Join' not exists {
        this exists
    }
}
rule filter_replication_source_identifier(reference_properties) {
    filter_is_not_valid_local_reference_via_join(%INPUT_DOCUMENT, %reference_properties, "AWS::RDS::DBCluster")
    filter_is_not_valid_local_reference_via_join(%INPUT_DOCUMENT, %reference_properties, "AWS::RDS::DBInstance")
}
rule filter_is_not_valid_local_reference_via_join(doc, reference_properties, referenced_resource_type) {
    %reference_properties {
        this not is_string
        this is_struct
        when this.'Fn::Join' exists {
            'Fn::Join' {
                when filter_list_contains_valid_local_reference(this[1], %doc, %referenced_resource_type) {
                    this not exists
                } this exists
            }
        }
        when this.'Fn::Join' not exists {
            this exists
        }
    }
}
rule filter_list_contains_valid_local_reference(list, doc, referenced_resource_type) {
    some %list.* {
        check_local_references(%doc, this, %referenced_resource_type)
    }
}
# Utility Rules
# rule is_cfn_template(doc) {
#    %doc {
#        AWSTemplateFormatVersion exists or Resources exists
#    }
#}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
rule check_local_references(doc, reference_properties, referenced_resource_type) {
%reference_properties {
'Fn::GetAtt' {
    query_for_resource(%doc, this[0], %referenced_resource_type)
    <<Local Stack reference was invalid>>
} or Ref {
    query_for_resource(%doc, this, %referenced_resource_type)
    <<Local Stack reference was invalid>>
}
}

rule query_for_resource(doc, resource_key, referenced_resource_type) {
    let referenced_resource = %doc.Resources[ keys == %resource_key ]
    %referenced_resource not empty
    %referenced_resource {
        Type == %referenced_resource_type
    }
}

CT.RDS.PR.16 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
  DBClusterSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: RDS Cluster secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "examplemasteruser"}''
        GenerateStringKey: password
        PasswordLength: 16
        ExcludeCharacters: '"@/\'
  DBCluster:
    Type: AWS::RDS::DBCluster
    Properties:
      Engine: aurora-mysql
      MasterUsername:
        Fn::Sub: '{{resolve:secretsmanager:${DBClusterSecret}::username}}''
      MasterUserPassword:
        Fn::Sub: '{{resolve:secretsmanager:${DBClusterSecret}::password}}''
      StorageEncrypted: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  DBClusterSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: RDS Cluster secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "examplemasteruser"}''
        GenerateStringKey: password
        PasswordLength: 16
        ExcludeCharacters: '"@/"
[CT.RDS.PR.17] Require an Amazon RDS event notification subscription to have critical database instance events configured

This control checks whether your Amazon RDS event subscriptions for RDS instances are configured to notify on event categories of maintenance, failure, and configuration change.

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::EventSubscription
- **AWS CloudFormation guard rule:** [CT.RDS.PR.17 rule specification (p. 1158)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.17 rule specification (p. 1158)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.17 example templates (p. 1160)]

**Explanation**

Amazon RDS event notifications use Amazon SNS to make you aware of changes in the availability or configuration of your RDS resources. These notifications allow for rapid response.

**Usage considerations**

- This control applies only to Amazon RDS Event Subscriptions for RDS instances (SourceType of db-instance).

**Remediation for rule failure**

When SourceType is set to db-instance, set Enabled to true and ensure that the parameter EventCategories contains maintenance, failure, and configuration change values.

The examples that follow show how to implement this remediation.

**Amazon RDS Event Subscription - Example One**

Amazon RDS Event Subscription for RDS instances configured to notify on all event categories. The example is shown in JSON and in YAML.

**JSON example**
The examples that follow show how to implement this remediation.

**Amazon RDS Event Subscription - Example Two**

Amazon RDS Event Subscription for RDS instances configured to notify on maintenance, failure, and configuration change event categories. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "RDSEventSubscription": {
      "Type": "AWS::RDS::EventSubscription",
      "Properties": {
         "SnsTopicArn": {
            "Ref": "SnsTopic"
         },
         "SourceType": "db-instance",
         "Enabled": true
      }
   }
}
```

**YAML example**

```yaml
RDSEventSubscription:
   Type: AWS::RDS::EventSubscription
   Properties:
      SnsTopicArn: !Ref 'SnsTopic'
      SourceType: db-instance
      Enabled: true
```

1157
Proactive controls

EventCategories:
- maintenance
- failure
- configuration change

SourceType: db-instance

Enabled: true

CT.RDS.PR.17 rule specification

# # ###################################################################
##       Rule Specification        ##
# ###################################################################
#
# Rule Identifier:
#   rds_instance_event_notifications_configured_check
#
# Description:
#   Checks whether Amazon RDS event subscriptions for RDS instances are configured to
#   notify on event categories of 'maintenance', 'failure', and 'configuration change'.
#
# Reports on:
#   AWS::RDS::EventSubscription
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#     And: The input document does not contain any Amazon RDS event subscription Resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#            And: The input document contains an Amazon RDS event subscription resource
#            And: 'SourceType' is provided and is not 'db-instance'
#     Then: SKIP
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#            And: The input document contains an Amazon RDS event subscription resource
#            And: 'SourceType' is 'db-instance'
#            And: 'Enabled' is not provided or set to bool(false)
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#            And: The input document contains an Amazon RDS event subscription resource
#            And: 'SourceType' is provided and is 'db-instance'
#            And: 'Enabled' is provided and set to bool(true)
#            And: 'EventCategories' does not contain 'maintenance', 'failure', and 'configuration change'
#     Then: FAIL
#   Scenario: 5
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#            And: The input document contains an Amazon RDS event subscription resource
#            And: 'SourceType' is provided and is 'db-instance'
#            And: 'Enabled' is provided and set to bool(true)
#            And: 'EventCategories' does not exist or is an empty list
#     Then: PASS
# Scenario: 6
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon RDS event subscription resource
# And: 'SourceType' is provided and is 'db-instance'
# And: 'Enabled' is provided and set to bool(true)
# And: 'EventCategories' contains 'maintenance', 'failure', and 'configuration change'
# Then: PASS

# Constants

let RDS_EVENTSUBSCRIPTION_TYPE = "AWS::RDS::EventSubscription"
let INPUT_DOCUMENT = this
let EVENT_CATEGORIES = ["maintenance","failure","configuration change"]
let EVENT_SOURCE_TYPE = "db-instance"

# Assignments

let rds_event_subscriptions = Resources.*[ Type == %RDS_EVENTSUBSCRIPTION_TYPE ]

# Primary Rules

rule rds_instance_event_notifications_configured_check when
  is_cfn_template(%INPUT_DOCUMENT)
  %rds_event_subscriptions not empty {
    check(%rds_event_subscriptions.Properties)
<<
[CT.RDS.PR.17]: Require an Amazon RDS event notification subscription to have critical database instance events configured
[FIX]: When 'SourceType' is set to 'db-instance', set 'Enabled' to true and ensure that the parameter 'EventCategories' contains 'maintenance', 'failure', and 'configuration change' values.
>>
}

rule rds_instance_event_notifications_configured_check when is_cfn_hook(%INPUT_DOCUMENT,
  %RDS_EVENTSUBSCRIPTION_TYPE) {
  check(%INPUT_DOCUMENT.%RDS_EVENTSUBSCRIPTION_TYPE.resourceProperties)
<<
[CT.RDS.PR.17]: Require an Amazon RDS event notification subscription to have critical database instance events configured
[FIX]: When 'SourceType' is set to 'db-instance', set 'Enabled' to true and ensure that the parameter 'EventCategories' contains 'maintenance', 'failure', and 'configuration change' values.
>>
}

# Parameterized Rules

# check(resource) {
  %resource [ SourceType == %EVENT_SOURCE_TYPE ] {
    Enabled exists
    # Scenario 4
    Enabled == true
    # Scenario 5
    EventCategories not exists or
    # Scenario 6
    check_event_categories_for_required_events(EventCategories)
  }
}
rule check_event_categories_for_required_events(event_categories) {
    %event_categories {
        this exists
        this is_list
        this empty or
        %EVENT_CATEGORIES.* in this
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.RDS.PR.17 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
SNSTopic:
    Type: AWS::SNS::Topic
    Properties: {}
RDSEventSubscription:
    Type: AWS::RDS::EventSubscription
    Properties:
        SnsTopicArn:
            Ref: SNSTopic
        SourceType: db-instance
        Enabled: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
SNSTopic:
    Type: AWS::SNS::Topic
    Properties: {}
RDSEventSubscription:
    Type: AWS::RDS::EventSubscription
    Properties:
        SnsTopicArn:
            Ref: SNSTopic
        EventCategories:
            - maintenance
            - failure
        SourceType: db-instance
        Enabled: true
[CT.RDS.PR.18] Require an Amazon RDS event notification subscription to have critical database parameter group events configured

This control checks whether your Amazon RDS event subscriptions for RDS parameter groups are configured to notify on event categories of configuration change.

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::EventSubscription
- **AWS CloudFormation guard rule:** [CT.RDS.PR.18 rule specification (p. 1162)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.18 rule specification (p. 1162)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.18 example templates (p. 1165)]

**Explanation**

Amazon RDS event notifications use Amazon SNS to make you aware of changes in the availability or configuration of your RDS resources. These notifications allow for rapid response.

**Usage considerations**

- This control applies only to Amazon RDS event subscriptions for RDS parameter groups (SourceType of db-parameter-group).

**Remediation for rule failure**

When SourceType is set to db-parameter-group, set Enabled to true and ensure that the parameter EventCategories contains configuration change as a value.

The examples that follow show how to implement this remediation.

**Amazon RDS Event Subscription - Example One**

Amazon RDS event subscription for RDS parameter groups configured to notify on all event categories. The example is shown in JSON and in YAML.

**JSON example**

```
{
    "RDSEventSubscription": {
        "Type": "AWS::RDS::EventSubscription",
        "Properties": {
            "SnsTopicArn": {
                "Ref": "SnsTopic"
            },
            "SourceType": "db-parameter-group",
            "Enabled": true
        }
    }
}
```
The examples that follow show how to implement this remediation.

**Amazon RDS Event Subscription - Example Two**

Amazon RDS event subscription for RDS parameter groups configured to notify on the configuration change event category. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "RDSEventSubscription": {
    "Type": "AWS::RDS::EventSubscription",
    "Properties": {
      "SnsTopicArn": "!Ref 'SnsTopic'",
      "EventCategories": [
        "configuration change"
      ],
      "SourceType": "db-parameter-group",
      "Enabled": true
    }
  }
}
```

**YAML example**

```yaml
RDSEventSubscription:
  Type: AWS::RDS::EventSubscription
  Properties:
    SnsTopicArn: !Ref 'SnsTopic'
    EventCategories:
      - configuration change
    SourceType: db-parameter-group
    Enabled: true
```

**CT.RDS.PR.18 rule specification**

```
# ###################################
##       Rule Specification        ##
```

1162
# Rule Identifier:
# rds_pg_event_notifications_configured_check

# Description:
# Checks whether Amazon RDS event subscriptions for RDS parameter groups are configured
to notify on event categories of 'configuration change'.

# Reports on:
# AWS::RDS::EventSubscription

# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document does not contain any RDS event subscription resources
# Then: SKIP

# Scenario: 2
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an RDS event subscription resource
# And: 'SourceType' is provided and is not 'db-parameter-group'
# Then: SKIP

# Scenario: 3
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an RDS event subscription resource
# And: 'SourceType' is 'db-parameter-group'
# And: 'Enabled' is not provided or set to bool(false)
# Then: FAIL

# Scenario: 4
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an RDS event subscription resource
# And: 'SourceType' is provided and is 'db-parameter-group'
# And: 'Enabled' is provided and set to bool(true)
# And: 'EventCategories' does not contain 'configuration change'
# Then: FAIL

# Scenario: 5
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an RDS event subscription resource
# And: 'SourceType' is provided and is 'db-parameter-group'
# And: 'Enabled' is provided and set to bool(true)
# And: 'EventCategories' does not exist or is an empty list
# Then: PASS

# Scenario: 6
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an RDS event subscription resource
# And: 'SourceType' is provided and is 'db-parameter-group'
# And: 'Enabled' is provided and set to bool(true)
# And: 'EventCategories' contains 'configuration change'
# Then: PASS

# Constants
#
let RDS_EVENTSUBSCRIPTION_TYPE = "AWS::RDS::EventSubscription"
let INPUT_DOCUMENT = this
let EVENT_CATEGORIES = ["configuration change"]
let EVENT_SOURCE_TYPE = "db-parameter-group"
let rds_event_subscriptions = Resources.*[Type == %RDS_EVENTSUBSCRIPTION_TYPE]

# Primary Rules

rule rds_pg_event_notifications_configured_check when is_cfn_template(%INPUT_DOCUMENT)
   %rds_event_subscriptions not empty {
   check(%rds_event_subscriptions.Properties)
   <<
   [CT.RDS.PR.18]: Require an Amazon RDS event notification subscription to have critical database parameter group events configured
   [FIX]: When 'SourceType' is set to 'db-parameter-group', set 'Enabled' to true and ensure that the parameter 'EventCategories' contains 'configuration change' as a value.
   >>
}

rule rds_pg_event_notifications_configured_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_EVENTSUBSCRIPTION_TYPE) {
   check(%INPUT_DOCUMENT.%RDS_EVENTSUBSCRIPTION_TYPE.resourceProperties)
   <<
   [CT.RDS.PR.18]: Require an Amazon RDS event notification subscription to have critical database parameter group events configured
   [FIX]: When 'SourceType' is set to 'db-parameter-group', set 'Enabled' to true and ensure that the parameter 'EventCategories' contains 'configuration change' as a value.
   >>
}

# Parameterized Rules

rule check(resource) {
   %resource [SourceType == %EVENT_SOURCE_TYPE] {
   Enabled exists
   # Scenario 4
   Enabled == true
   # Scenario 5
   EventCategories not exists or
   # Scenario 6
   check_event_categories_for_required_events(EventCategories)
   }
}

rule check_event_categories_for_required_events(event_categories) {
   %event_categories {
   this exists
   this is_list
   this empty or
   %EVENT_CATEGORIES.* in this
   }
}

# Utility Rules

rule is_cfn_template(doc) {
   %doc {
   AWSTemplateFormatVersion exists or Resources exists
   }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
   %doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.RDS.PR.18 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
SNSTopic:
  Type: AWS::SNS::Topic
  Properties: {}
RDSEventSubscription:
  Type: AWS::RDS::EventSubscription
  Properties:
    SnsTopicArn:
      Ref: SNSTopic
    SourceType: db-parameter-group
    Enabled: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
SNSTopic:
  Type: AWS::SNS::Topic
  Properties: {}
RDSEventSubscription:
  Type: AWS::RDS::EventSubscription
  Properties:
    SnsTopicArn:
      Ref: SNSTopic
    Enabled: false
    SourceType: db-parameter-group

[CT.RDS.PR.19] Require an Amazon RDS event notifications subscription to have critical database security group events configured

This control checks whether your Amazon RDS event subscriptions for RDS security groups are configured to notify on event categories of failure and configuration change

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::EventSubscription
- **AWS CloudFormation guard rule:** CT.RDS.PR.19 rule specification (p. 1167)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.RDS.PR.19 rule specification (p. 1167)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.RDS.PR.19 example templates (p. 1169)
Explanation

Amazon RDS event notifications use Amazon SNS to make you aware of changes in the availability or configuration of your RDS resources. These notifications allow for a rapid response.

Usage considerations

- This control applies only to Amazon RDS Event Subscriptions for RDS security groups (SourceType of db-security-group)

Remediation for rule failure

When SourceType is set to db-security-group, set Enabled to true and ensure that the parameter EventCategories contains both failure and configuration change values.

The examples that follow show how to implement this remediation.

Amazon RDS Event Subscription - Example One

Amazon RDS Event Subscription for RDS security groups configured to notify on all event categories. The example is shown in JSON and in YAML.

JSON example

```
{
    "RDSEventSubscription": {
        "Type": "AWS::RDS::EventSubscription",
        "Properties": {
            "SnsTopicArn": {
                "Ref": "SnsTopic"
            },
            "SourceType": "db-security-group",
            "Enabled": true
        }
    }
}
```

YAML example

```
RDSEventSubscription:
  Type: AWS::RDS::EventSubscription
  Properties:
    SnsTopicArn: !Ref 'SnsTopic'
    SourceType: db-security-group
    Enabled: true
```

The examples that follow show how to implement this remediation.

Amazon RDS Event Subscription - Example Two

Amazon RDS Event Subscription for RDS security groups configured to notify on failure and configuration change event categories. The example is shown in JSON and in YAML.

JSON example

```
{
    "RDSEventSubscription": {
        "Type": "AWS::RDS::EventSubscription",
        "Properties": {
            "SnsTopicArn": {
                "Ref": "SnsTopic"
            },
            "SourceType": "db-security-group",
            "Enabled": true
        }
    }
}
```

YAML example

```
RDSEventSubscription:
  Type: AWS::RDS::EventSubscription
  Properties:
    SnsTopicArn: !Ref 'SnsTopic'
    SourceType: db-security-group
    Enabled: true
```
YAML example

RDSEventSubscription:
  Type: AWS::RDS::EventSubscription
  Properties:
    SnsTopicArn: !Ref 'SnsTopic'
    EventCategories:
      - failure
      - configuration change
    SourceType: db-security-group
    Enabled: true

CT.RDS.PR.19 rule specification

# ****************************
#   Rule Specification      
# ****************************
#
# Rule Identifier:
#   rds_sg_event_notifications_configured_check
# Description:
#   Checks whether an Amazon RDS event subscription for RDS security groups are configured
to notify on event categories of 'failure' and 'configuration change'.
# Reports on:
#   AWS::RDS::EventSubscription
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
#   None
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#        And: The input document does not contain any Amazon RDS event subscription
#        resources
#     Then: SKIP
#   Scenario: 2
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an Amazon RDS event subscription Resource
And: 'SourceType' is provided and is not 'db-security-group'
Then: SKIP

Scenario: 3
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an Amazon RDS event subscription resource
And: 'SourceType' is 'db-security-group'
And: 'Enabled' is not provided or set to bool(false)
Then: FAIL

Scenario: 4
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an Amazon RDS event subscription resource
And: 'SourceType' is 'db-security-group'
And: 'Enabled' is provided and set to bool(true)
And: 'EventCategories' does not contain both 'failure' and 'configuration change'
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an Amazon RDS event subscription resource
And: 'SourceType' is provided and is 'db-security-group'
And: 'Enabled' is provided and set to bool(true)
And: 'EventCategories' does not exist or is an empty list
Then: PASS

Scenario: 6
Given: The input document is an AWS CloudFormation or CloudFormation hook document
And: The input document contains an Amazon RDS event subscription resource
And: 'SourceType' is provided and is 'db-security-group'
And: 'Enabled' is provided and set to bool(true)
And: 'EventCategories' contains both 'failure' and 'configuration change'
Then: PASS

# Constants

let RDS_EVENTSUBSCRIPTION_TYPE = "AWS::RDS::EventSubscription"
let INPUT_DOCUMENT = this
let EVENT_CATEGORIES = ["failure","configuration change"]
let EVENT_SOURCE_TYPE = "db-security-group"

# Assignments

let rds_event_subscriptions = Resources.*[ Type == %RDS_EVENTSUBSCRIPTION_TYPE ]

# Primary Rules

rule rds_sg_event_notifications_configured_check when is_cfn_template(%INPUT_DOCUMENT)
%rds_event_subscriptions not empty  {
    check(%rds_event_subscriptions.Properties)
    <<
    [CT.RDS.PR.19]: Require an Amazon RDS event notifications subscription to have
critical database security group events configured
    [FIX]: When 'SourceType' is set to 'db-security-group', set 'Enabled' to true and
ensure that the parameter 'EventCategories' contains both 'failure' and 'configuration
change' values.
    >>
}

rule rds_sg_event_notifications_configured_check when is_cfn_hook(%INPUT_DOCUMENT,
%RDS_EVENTSUBSCRIPTION_TYPE) {
    check(%INPUT_DOCUMENT.%RDS_EVENTSUBSCRIPTION_TYPE.resourceProperties)
    <<
    [CT.RDS.PR.19]: Require an Amazon RDS event notifications subscription to have
critical database security group events configured
[FIX]: When 'SourceType' is set to 'db-security-group', set 'Enabled' to true and ensure that the parameter 'EventCategories' contains both 'failure' and 'configuration change' values.

```plaintext
# Parameterized Rules
# rule check(resource) {
    %resource [SourceType == %EVENT_SOURCE_TYPE] {
        Enabled exists
        # Scenario 4
        Enabled == true
        # Scenario 5
        EventCategories not exists or
        # Scenario 6
        check_event_categories_for_required_events(EventCategories)
    }
}

rule check_event_categories_for_required_events(event_categories) {
    %event_categories {
        this exists
        this is_list
        this empty or
        %EVENT_CATEGORIES.* in this
    }
}

# Utility Rules
# rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

CT.RDS.PR.19 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
- **SNSTopic**: Type: AWS::SNS::Topic
  Properties: {}
- **RDSEventSubscription**: Type: AWS::RDS::EventSubscription
  Properties:
    SnsTopicArn:
      Ref: SNSTopic
    SourceType: db-security-group
    Enabled: true
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
SNSTopic:
  Type: AWS::SNS::Topic
  Properties: {}
RDSEventSubscription:
  Type: AWS::RDS::EventSubscription
  Properties:
    SnsTopicArn:
      Ref: SNSTopic
    EventCategories:
      - failure
    SourceType: db-security-group
    Enabled: true

[CT.RDS.PR.20] Require an Amazon RDS database instance not to use a database engine default port

This control checks whether Amazon Relational Database Service (RDS) database instances are configured for default database port for their specific engine types.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBInstance
- **AWS CloudFormation guard rule:** [CT.RDS.PR.20 rule specification](p. 1172)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.20 rule specification](p. 1172)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.20 example templates](p. 1175)

Explanation

If you use a known port to deploy an Amazon RDS cluster or instance, an attacker can guess information about the cluster or instance. The attacker can use this information in conjunction with other information to connect to an Amazon RDS cluster or instance, or to gain additional information about your application.

When you change the port, you must also update the existing connection strings that were used to connect to the old port. You also should check the security group of the DB instance to ensure that it includes an ingress rule that allows connectivity on the new port.

Usage considerations

- This control applies only to Amazon RDS DB engine types mariadb, mysql, oracle-ee, oracle-se2, oracle-ee-cdb, oracle-se2-cdb, postgres, sqlserver-ee, sqlserver-se, sqlserver-ex and sqlserver-web.
Remediation for rule failure

Set a value for Port that is different than the default value for the Amazon RDS DB instance engine type.

The examples that follow show how to implement this remediation.

Amazon RDS DB Instance - Example One

Amazon RDS DB instance configured with a port that's different than the mysql engine default port. The example is shown in JSON and in YAML.

JSON example

```
{
  "DBInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "Engine": "mysql",
      "EngineVersion": 5.7,
      "DBInstanceClass": "db.m5.large",
      "StorageType": "gp2",
      "AllocatedStorage": 5,
      "MasterUsername": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}"
      },
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
      },
      "StorageEncrypted": true,
      "Port": 6733
    },
    "DeletionPolicy": "Delete"
  }
}
```

YAML example

```
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: mysql
    EngineVersion: 5.7
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    StorageEncrypted: true
    Port: 6733
    DeletionPolicy: Delete
```

The examples that follow show how to implement this remediation.

Amazon RDS DB Instance - Example Two

Amazon RDS DB instance configured with a port that's different than the postgres engine default port. The example is shown in JSON and in YAML.
JSON example

```json
{
  "DBInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "Engine": "postgres",
      "EngineVersion": "14.2",
      "DBInstanceClass": "db.m5.large",
      "StorageType": "gp2",
      "AllocatedStorage": 5,
      "MasterUsername": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}"
      },
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
      },
      "Port": 5723
    },
    "DeletionPolicy": "Delete"
  }
}
```

YAML example

```yaml
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    Port: 5723
  DeletionPolicy: Delete
```

CT.RDS.PR.20 rule specification

```bash
# #######################################################################
##       Rule Specification       ##
# ############################################################################
#
# Rule Identifier:
#   rds_instance_no_default_ports_check
#
# Description:
#   This control checks whether Amazon Relational Database Service (RDS) database instances
#   are configured for default database port for their specific engine types.
#
# Reports on:
#   AWS::RDS::DBInstance
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
```
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#  And: The input document does not contain any Amazon RDS DB instance resources
#  Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#  And: The input document contains an RDS DB instance resource
#  And: 'Engine' is not one of 'mariadb', 'mysql',
#       'oracle-ee', 'oracle-se2', 'oracle-ee-cdb', 'oracle-se2-cdb',
#       'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#  Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#  And: The input document contains an RDS DB instance resource
#  And: 'Engine' is one of 'mariadb', 'mysql',
#       'oracle-ee', 'oracle-se2', 'oracle-ee-cdb', 'oracle-se2-cdb',
#       'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#  And: 'Port' has not been specified
#  Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#  And: The input document contains an RDS DB instance resource
#  And: 'Engine' is one of 'mariadb', 'mysql',
#       'oracle-ee', 'oracle-se2', 'oracle-ee-cdb', 'oracle-se2-cdb',
#       'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#  And: 'Port' has been specified
#  And: 'Port' value is default port (includes mysql/mariadb port '3306', sqlserver port '1433', postgres port '5432', and oracle port '1521')
#  Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#  And: The input document contains an RDS DB instance resource
#  And: 'Engine' is one of 'mariadb', 'mysql'
#  And: 'Port' has been specified
#  And: 'Port' value is not equal to '3306'
#  Then: PASS
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#  And: The input document contains an RDS DB instance resource
#  And: 'Engine' is 'postgres'
#  And: 'Port' has been specified
#  And: 'Port' value is not equal to '5432'
#  Then: PASS
# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#  And: The input document contains an RDS DB instance resource
#  And: 'Engine' is one of 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex', 'sqlserver-web'
#  And: 'Port' has been specified
#  And: 'Port' value is not equal to '1433'
#  Then: PASS
# Scenario: 8
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#  And: The input document contains an RDS DB instance resource
And: 'Engine' is one of 'oracle-ee', 'oracle-se2', 'oracle-ee-cdb', 'oracle-se2-cdb',
And: 'Port' has been specified
And: 'Port' value is not equal to '1521'
Then: PASS

# Constants

let RDS_DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let INPUT_DOCUMENT = this
let ORACLE_ENGINES = ["oracle-ee", "oracle-se2", "oracle-se1", "oracle-se"]
let SQLSERVER_ENGINES = ["sqlserver-ee", "sqlserver-se", "sqlserver-ex", "sqlserver-web"]
let MYSQL_OR_MARIA_ENGINES = ["mariadb", "mysql"]
let POSTGRES_ENGINES = ["postgres"]
let MYSQL_MARIA_DEFAULT_PORTS = [3306, "3306"]
let POSTGRES_DEFAULT_PORTS = [5432, "5432"]
let SQL_DEFAULT_PORTS = [1433, "1433"]
let ORACLE_DEFAULT_PORTS = [1521, "1521"]

# Assignments

let rds_db_instances = Resources.*[ Type == %RDS_DB_INSTANCE_TYPE ]

# Primary Rules

# rule rds_instance_no_default_ports_check when is_cfn_template(%INPUT_DOCUMENT)

%rds_db_instances not empty {
  check(%rds_db_instances.Properties)
  [CT.RDS.PR.20]: Require an Amazon RDS database instance not to use a database
  engine default port
  [FIX]: Set a value for 'Port' that is different than the default value for the
  Amazon RDS DB instance engine type.
}

rule rds_instance_no_default_ports_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_INSTANCE_TYPE) {
  check(%INPUT_DOCUMENT.%RDS_DB_INSTANCE_TYPE.resourceProperties)
  [CT.RDS.PR.20]: Require an Amazon RDS database instance not to use a database
  engine default port
  [FIX]: Set a value for 'Port' that is different than the default value for the
  Amazon RDS DB instance engine type.
}

# Parameterized Rules

# rule check(rds_db_instance) {
  # Scenario: 4 and 5
  %rds_db_instance[ filter_engine(this, %MYSQL_OR_MARIA_ENGINES) ] {
    check_port(Port, %MYSQL_MARIA_DEFAULT_PORTS)
  }
  # Scenario: 4 and 6
  %rds_db_instance[ filter_engine(this, %POSTGRES_ENGINES) ] {
    check_port(Port, %POSTGRES_DEFAULT_PORTS)
  }
  # Scenario: 4 and 7
  %rds_db_instance[ filter_engine(this, %SQLSERVER_ENGINES) ] {
    check_port(Port, %SQL_DEFAULT_PORTS)
  }
}
# Scenario: 4 and 8
%rds_db_instance[ filter_engine(this, %ORACLE_ENGINES) ] {  
    check_port(Port, %ORACLE_DEFAULT_PORTS)  
}

rule filter_engine(db_properties, engine) {  
    %db_properties {  
        # Scenario: 2  
        Engine exists  
        Engine is_string  
        Engine in %engine  
    }  
}

rule check_port(port, default_ports) {  
    # Scenario: 3  
    %port exists  
    %port not in %default_ports  
}

# Utility Rules
#
rule is_cfn_template(doc) {  
    %doc {  
        AWSTemplateFormatVersion exists or  
        Resources exists  
    }  
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {  
    %doc.%RESOURCE_TYPE.resourceProperties exists  
}

CT.RDS.PR.20 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
DBInstanceSecret:
    Type: AWS::SecretsManager::Secret
Properties:
    Description: RDS DB instance secret
    GenerateSecretString:
        SecretStringTemplate: '{"username": "testUser"}'
        GenerateStringKey: password
        PasswordLength: 22
        ExcludeCharacters: '"/@\'
DBInstance:
    Type: AWS::RDS::DBInstance
Properties:
    Engine: mysql
    EngineVersion: 5.7
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername:
        Fn::Sub: '${resolve:secretsmanager:${DBInstanceSecret}::username}'}
[CT.RDS.PR.21] Require an Amazon RDS DB cluster to have a unique administrator username

This control checks whether an Amazon Relational Database Service (RDS) database (DB) cluster has changed the administrator username from its default value.

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBCluster
- **AWS CloudFormation guard rule:** [CT.RDS.PR.21 rule specification (p. 1177)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.21 rule specification (p. 1177)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.21 example templates (p. 1179)]

Explanation
When you create an Amazon RDS database, we recommend that you change the default administrator username to a unique value. Default user names are public knowledge, and they should be changed, because changing these user names reduces the risk of unintended access.

**Usage considerations**

- This control applies only to Amazon RDS DB clusters that set the `MasterUsername` property.

**Remediation for rule failure**

Set `MasterUsername` to a value other than `admin` or `postgres`.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Cluster - Example**

Amazon RDS DB cluster configured with a unique administrator username. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "DBCluster": {
    "Type": "AWS::RDS::DBCluster",
    "Properties": {
      "Engine": "aurora-mysql",
      "MasterUsername": "samplemasteruser",
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${RDSClusterSecret}::password}}"
      },
      "DBSubnetGroupName": {
        "Ref": "DBSubnetGroup"
      }
    }
  }
}
```

**YAML example**

```yaml
DBCluster:
  Type: AWS::RDS::DBCluster
  Properties:
    Engine: aurora-mysql
    MasterUsername: samplemasteruser
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${RDSClusterSecret}::password}}'
    DBSubnetGroupName: !Ref 'DBSubnetGroup'
```

**CT.RDS.PR.21 rule specification**

```yaml
# ###################################
##       Rule Specification        ##
#####################################
```
# Rule Identifier:
#   rds_cluster_default_admin_check
#
# Description:
#   This control checks whether an Amazon Relational Database Service (RDS) database (DB)
#   cluster has changed the administrator username from its default value.
#
# Reports on:
#   AWS::RDS::DBCluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#     And: The input document does not contain any RDS DB cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document contains an RDS DB cluster resource
#           And: 'MasterUsername' has not been provided
#     Then: SKIP
#   Scenario: 3
#     Given: The input document contains an RDS DB cluster resource
#           And: 'MasterUsername' has been provided and it is set to 'admin' or 'postgres'
#     Then: FAIL
#   Scenario: 4
#     Given: The input document contains an RDS DB cluster resource
#           And: 'MasterUsername' has been provided and is not set to 'admin' or 'postgres'
#     Then: PASS
#
# Constants
#
let RDS_DB_CLUSTER_TYPE = "AWS::RDS::DBCluster"
let DISALLOWED_MASTER_USERNAMES = ["admin", "postgres"]
let INPUT_DOCUMENT = this
#
# Assignments
#
let db_clusters = Resources.*[ Type == %RDS_DB_CLUSTER_TYPE ]
#
# Primary Rules
#
rule rds_cluster_default_admin_check when is_cfn_template(%INPUT_DOCUMENT)
  %db_clusters not empty { 
    check(%db_clusters.Properties)
    <<
      [CT.RDS.PR.21]: Require an Amazon RDS DB cluster to have a unique administrator
      username
      [FIX]: Set 'MasterUsername' to a value other than 'admin' or 'postgres'.
    >>
  }
rule rds_cluster_default_admin_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_CLUSTER_TYPE) {
    check(%INPUT_DOCUMENT.%RDS_DB_CLUSTER_TYPE.resourceProperties)
    "<<
    [CT.RDS.PR.21]: Require an Amazon RDS DB cluster to have a unique administrator username
    [FIX]: Set 'MasterUsername' to a value other than 'admin' or 'postgres'.
    >>
}

rule check(db_cluster) {
    %db_cluster [
        # scenario 2
        filter_master_username_provided(this)
    ]{
        # scenario 3 and 4
        MasterUsername not in %DISALLOWED_MASTER_USERNAMES
    }
}

# Utility Rules
#
# Utility Rules
#
rule filter_master_username_provided(dbcluster_properties) {
    %dbcluster_properties{
        MasterUsername exists
    }
}

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.RDS.PR.21 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
VPC:
    Type: AWS::EC2::VPC
    Properties:
        CidrBlock: 10.0.0.0/16
        EnableDnsHostnames: true
        EnableDnsSupport: true
SubnetOne:
    Type: AWS::EC2::Subnet
    Properties:
        VpcId:
            Ref: VPC
        CidrBlock: 10.0.0.0/25
        AvailabilityZone:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.
[CT.RDS.PR.22] Require an Amazon RDS database instance to have a unique administrator username

This control checks whether an Amazon Relational Database Service (RDS) database has changed the administrator username from its default value.

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBInstance
- **AWS CloudFormation guard rule:** [CT.RDS.PR.22 rule specification (p. 1182)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.22 rule specification (p. 1182)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.22 example templates (p. 1185)]

**Explanation**

Default administrative usernames on Amazon RDS databases are public knowledge. When creating an Amazon RDS database, you should change the default administrative username to a unique value, thereby reducing the risk of unintended access.
Usage considerations

- This control applies only to Amazon RDS DB engine types mariadb, mysql, oracle-ee, oracle-ee-cdb, oracle-se2, oracle-se2-cdb, postgres, sqlserver-ee, sqlserver-se, sqlserver-ex, and sqlserver-web.

Remediation for rule failure

Set MasterUsername to a value other than postgres or admin.

The examples that follow show how to implement this remediation.

Amazon RDS DB Instance - Example

Amazon RDS DB instance configured with a custom administrator username. The example is shown in JSON and in YAML.

JSON example

```
{
  "DBInstance": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "Engine": "postgres",
      "EngineVersion": 14.2,
      "DBInstanceClass": "db.m5.large",
      "StorageType": "gp2",
      "AllocatedStorage": 5,
      "MasterUsername": "testUser",
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
      }
    },
    "DeletionPolicy": "Delete"
  }
}
```

YAML example

```
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: testUser
    MasterUserPassword: !Sub "{{resolve:secretsmanager:${DBInstanceSecret}::password}}"
  DeletionPolicy: Delete
```

CT.RDS.PR.22 rule specification

```
# ##########################################################################
##       Rule Specification        ##
1182
```
# Rule Identifier:
# rds_instance_default_admin_check
#
# Description:
# This control checks whether an Amazon Relational Database Service (RDS) database has
# changed the administrator username from its default value.
#
# Reports on:
# AWS::RDS::DBInstance
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document does not contain any RDS DB instance resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is not one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb',
# 'oracle-se2',
# 'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
# 'sqlserver-web'
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-
# se2',
# 'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
# 'sqlserver-web'
# And: 'MasterUsername' has been specified and is one of 'postgres' or 'admin'
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-
# se2',
# 'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
# 'sqlserver-web'
# And: 'MasterUsername' has been specified and is not one of 'postgres' or 'admin'
# Then: PASS
#
# Constants
#
let RDS_DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let INPUT_DOCUMENT = this
let SUPPORTED_RDS_INSTANCE_ENGINES = [
  "mariadb", "mysql", "oracle-ee", "oracle-ee-cdb", "oracle-se2",
  "oracle-se2-cdb", "postgres", "sqlserver-ee", "sqlserver-se",
  "sqlserver-ex", "sqlserver-web"
]
let RDS_DEFAULT_USERNAMES = [ "postgres", "admin" ]
# Assignments

let rds_db_instances = Resources.*[ Type == %RDS_DB_INSTANCE_TYPE ]

# Primary Rules

rule rds_instance_default_admin_check when is_cfn_template(%INPUT_DOCUMENT)
%rds_db_instances not empty {
  check(%rds_db_instances.Properties)
  <<
  [CT.RDS.PR.22]: Require an Amazon RDS database instance to have a unique administrator username
  [FIX]: Set 'MasterUsername' to a value other than 'postgres' or 'admin'.
  >>
}

rule rds_instance_default_admin_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_INSTANCE_TYPE) {
  check(%INPUT_DOCUMENT.%RDS_DB_INSTANCE_TYPE.resourceProperties)
  <<
  [CT.RDS.PR.22]: Require an Amazon RDS database instance to have a unique administrator username
  [FIX]: Set 'MasterUsername' to a value other than 'postgres' or 'admin'.
  >>
}

# Parameterized Rules

rule check(rds_db_instance) {
  %rds_db_instance [ filter_engine_and_master_username_provided(this) ] {
    # Scenario: 3 and 4
    MasterUsername not in %RDS_DEFAULT_USERNAMES
  }
}

rule filter_engine_and_master_username_provided(db_properties) {
  %db_properties {
    # Scenario: 2
    MasterUsername exists
    Engine exists
    Engine is_string
    Engine in %SUPPORTED_RDS_INSTANCE_ENGINES
  }
}

# Utility Rules

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.RDS.PR.22 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  DBInstanceSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: Test RDS DB Instance secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "testUser"}'
        GenerateStringValue: password
        PasswordLength: 22
        ExcludeCharacters: '@\'
  DBInstance:
    Type: AWS::RDS::DBInstance
    Properties:
      Engine: postgres
      EngineVersion: 14.2
      DBInstanceClass: db.m5.large
      StorageType: gp2
      AllocatedStorage: 5
      MasterUsername: testUser
      MasterUserPassword:
        Fn::Sub: '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    DeletionPolicy: Delete
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
  DBInstanceSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: Test RDS DB Instance secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "testUser"}'
        GenerateStringValue: password
        PasswordLength: 22
        ExcludeCharacters: '@\'
  DBInstance:
    Type: AWS::RDS::DBInstance
    Properties:
      Engine: postgres
      EngineVersion: 14.2
      DBInstanceClass: db.m5.large
      StorageType: gp2
      AllocatedStorage: 5
      MasterUsername: postgres
      MasterUserPassword:
        Fn::Sub: '{{resolve:secretsmanager:${DBInstanceSecret}::password}}'
    DeletionPolicy: Delete
```
**[CT.RDS.PR.23] Require an Amazon RDS database instance to not be publicly accessible**

This rule checks whether Amazon Relational Database Service (RDS) database (DB) instances are publicly accessible, as determined by checking the PubliclyAccessible configuration property.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBInstance
- **AWS CloudFormation guard rule:** [CT.RDS.PR.23 rule specification (p. 1187)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.23 rule specification (p. 1187)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.23 example templates (p. 1189)]

**Explanation**

The PubliclyAccessible property in the RDS instance CloudFormation resource indicates whether the DB instance is publicly accessible. When the DB instance is configured with PubliclyAccessible set to true, it is an internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. When the DB instance isn't publicly accessible, it is an internal instance with a DNS name that resolves to a private IP address.

Unless you intend for your RDS instance to be publicly accessible, do not configure the RDS instance with the PubliclyAccessible value set to true, because this configuration may allow unwanted traffic to your database instance.

**Remediation for rule failure**

Set the value of PubliclyAccessible to false.

The examples that follow show how to implement this remediation.

**Amazon RDS DB Instance - Example**

Amazon RDS DB instance configured as an internal instance, by means of a publicly accessible configuration. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "DBInstance": {
      "Type": "AWS::RDS::DBInstance",
      "Properties": {
         "Engine": "postgres",
         "EngineVersion": "14.2",
         "DBInstanceClass": "db.m5.large",
         "StorageType": "gp2",
         "AllocatedStorage": 5,
         "MasterUsername": {
            "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::username}}"
         }
      }
   }
}
```
"MasterUserPassword": {
    "Fn::Sub": "{{resolve:secretsmanager:${DBInstanceSecret}::password}}",
    "PubliclyAccessible": false
},
"DeletionPolicy": "Delete"
}

YAML example

DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{${resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword: !Sub '{${resolve:secretsmanager:${DBInstanceSecret}::password}}'
    PubliclyAccessible: false
    DeletionPolicy: Delete

CT.RDS.PR.23 rule specification

# ###############################################################################
##      Rule Specification      ##
# ###############################################################################

# Rule Identifier:
#   rds_instance_public_access_check
#
# Description:
#   This rule checks whether Amazon Relational Database Service (RDS) database (DB) instances are publicly accessible, as determined by checking the 'PubliclyAccessible' configuration property.
#
# Reports on:
#   AWS::RDS::DBInstance
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#      And: The input document does not contain any RDS DB instance resources
#      Then: SKIP
#   Scenario: 2
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#      And: The input document contains an RDS DB instance resource
#      And: 'PubliclyAccessible' has not been specified
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB instance resource
# And: 'PubliclyAccessible' is present and is a value other than bool(false)
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB instance resource
# And: 'PubliclyAccessible' has been specified and set to bool(false)
# Then: PASS

# Constants
let RDS_DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let INPUT_DOCUMENT = this

# Assignments
let rds_db_instances = Resources.*[ Type == %RDS_DB_INSTANCE_TYPE ]

# Primary Rules
# rule rds_instance_public_access_check when is_cfn_template(%INPUT_DOCUMENT)
%rds_db_instances not empty {
  %rds_db_instance{
    # Scenario: 2
    PubliclyAccessible exists
    # Scenario: 3 and 4
    PubliclyAccessible == false
  }
}

rule rds_instance_public_access_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_INSTANCE_TYPE) {
  %rds_db_instance{
    # Scenario: 2
    PubliclyAccessible exists
    # Scenario: 3 and 4
    PubliclyAccessible == false
  }
}

# Parameterized Rules
# rule check(rds_db_instance) {
  %rds_db_instance{
    # Scenario: 2
    PubliclyAccessible exists
    # Scenario: 3 and 4
    PubliclyAccessible == false
  }
}

# Utility Rules
# rule is_cfn_template(doc) {
  %doc {

ct.rds.pr.23 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
DBInstanceSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: RDS DB instance secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "examplemasteruser"}'
      GenerateStringKey: password
      PasswordLength: 22
      ExcludeCharacters: '"@/\'
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername:
      Fn::Sub: '{{$resolve:secretsmanager:${DBInstanceSecret}::username}}'
    MasterUserPassword:
      Fn::Sub: '{{$resolve:secretsmanager:${DBInstanceSecret}::password}}'
    PubliclyAccessible: false
    DeletionPolicy: Delete

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
DBInstanceSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: RDS DB instance secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "examplemasteruser"}'
      GenerateStringKey: password
      PasswordLength: 22
      ExcludeCharacters: '"@/\'
DBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
[CT.RDS.PR.24] Require an Amazon RDS database instance to have encryption at rest configured

This control checks whether storage encryption is enabled for your Amazon RDS database (DB) instance.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::RDS::DBInstance
- **AWS CloudFormation guard rule:** [CT.RDS.PR.24 rule specification](p. 1191)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.24 rule specification](p. 1191)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.24 example templates](p. 1193)

Explanation

For an added layer of security for your sensitive data in Amazon RDS DB instances, you should configure your RDS DB instances to be encrypted at rest. To encrypt your RDS DB instances and snapshots at rest, enable the encryption option for your RDS DB instances. Data that is encrypted at rest includes the underlying storage for DB instances, its automated backups, read replicas, and snapshots.

Encrypted Amazon RDS DB instances use the open standard AES-256 encryption algorithm to encrypt your data on the server that hosts your RDS DB instances. After your data is encrypted, Amazon RDS handles authentication of access and decryption of your data transparently with a minimal impact on performance. You do not need to modify your database client applications to use encryption.

Amazon RDS encryption currently is available for all database engines and storage types. Amazon RDS encryption is available for most DB instance classes.

**Usage considerations**

- This control applies only to Amazon RDS DB engine types mariadb, mysql, oracle-ee, oracle-ee-cdb, oracle-se2, oracle-se2-cdb, postgres, sqlserver-ee, sqlserver-se, sqlserver-ex and sqlserver-web

**Remediation for rule failure**

The parameter StorageEncrypted must be set to true for RDS DB Instances.

The examples that follow show how to implement this remediation.
Amazon RDS DB instance - Example

Amazon RDS DB instance with storage encryption enabled. The example is shown in JSON and in YAML.

**JSON example**

```
{
    "RDSDBInstance": {
        "Type": "AWS::RDS::DBInstance",
        "Properties": {
            "Engine": "postgres",
            "EngineVersion": 14.2,
            "DBInstanceClass": "db.m5.large",
            "StorageType": "gp2",
            "AllocatedStorage": 5,
            "MasterUsername": {
                "Fn::Sub": "{{resolve:secretsmanager:${RDSDBInstanceSecret}::username}}" 
            },
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager:${RDSDBInstanceSecret}::password}}" 
            },
            "StorageEncrypted": true
        }
    }
}
```

**YAML example**

```
RDSDBInstance:
  Type: AWS::RDS::DBInstance
  Properties:
    Engine: postgres
    EngineVersion: 14.2
    DBInstanceClass: db.m5.large
    StorageType: gp2
    AllocatedStorage: 5
    MasterUsername: !Sub '{{resolve:secretsmanager:${RDSDBInstanceSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${RDSDBInstanceSecret}::password}}'
    StorageEncrypted: true
```

**CT.RDS.PR.24 rule specification**

```
# #####################################################################
##       Rule Specification       ##
# #####################################################################
#
# Rule Identifier:
#   rds_instance_storage_encrypted_check
#
# Description:
#   Checks whether storage encryption is enabled for your Amazon RDS DB instances.
#
# Reports on:
#   AWS::RDS::DBInstance
#
# Evaluates:
```

```
1191
```
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document does not contain any RDS DB instance resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is not one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb',
#     'oracle-se2',
#     'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#     'sqlserver-web'
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-
#     se2',
#     'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#     'sqlserver-web'
# And: 'StorageEncrypted' has not been provided
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-
#     se2',
#     'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#     'sqlserver-web'
# And: 'StorageEncrypted' has been provided and set to bool(false)
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an RDS DB instance resource
# And: 'Engine' is one of 'mariadb', 'mysql', 'oracle-ee', 'oracle-ee-cdb', 'oracle-
#     se2',
#     'oracle-se2-cdb', 'postgres', 'sqlserver-ee', 'sqlserver-se', 'sqlserver-ex',
#     'sqlserver-web'
# And: 'StorageEncrypted' has been provided and set to bool(true)
# Then: PASS
#
# Constants
#
let RDS_DB_INSTANCE_TYPE = "AWS::RDS::DBInstance"
let INPUT_DOCUMENT = this
let SUPPORTED_RDS_INSTANCE_ENGINES = [
    "mariadb", "mysql", "oracle-ee", "oracle-ee-cdb", "oracle-se2",
    "oracle-se2-cdb", "postgres", "sqlserver-ee", "sqlserver-se",
    "sqlserver-ex", "sqlserver-web"
]
#
# Assignments
#
let rds_db_instances = Resources.*[ Type == %RDS_DB_INSTANCE_TYPE ]
#
# Primary Rules
#
rule rds_instance_storage_encrypted_check when is_cfn_template(%INPUT_DOCUMENT)
%rds_db_instances not empty {
  check(%rds_db_instances.Properties)
  <<
  [CT.RDS.PR.24]: Require an Amazon RDS database instance to have encryption at rest configured
  [FIX]: The parameter 'StorageEncrypted' must be set to true for RDS DB Instances.
  >>
}

rule rds_instance_storage_encrypted_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_INSTANCE_TYPE) {
  check(%INPUT_DOCUMENT.%RDS_DB_INSTANCE_TYPE.resourceProperties)
  <<
  [CT.RDS.PR.24]: Require an Amazon RDS database instance to have encryption at rest configured
  [FIX]: The parameter 'StorageEncrypted' must be set to true for RDS DB Instances.
  >>
}

# Parameterized Rules

rule check(rds_db_instance) {
  %rds_db_instance [filter_restore_and_engine(this)] {
    #Scenario: 3
    StorageEncrypted exists
    #Scenario: 4 and 5
    StorageEncrypted == true
  }
}

rule filter_restore_and_engine(db_properties) {
  %db_properties {
    #Scenario: 2
    Engine exists
    Engine is_string
    Engine in %SUPPORTED_RDS_INSTANCE_ENGINES
  }
}

# Utility Rules

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.RDS.PR.24 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:

1193
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

[CT.RDS.PR.25] Require an Amazon RDS database cluster to have logging configured

This control checks whether Amazon RDS database clusters have all available log types enabled for export to Amazon CloudWatch Logs.

- Control objective: Establish logging and monitoring
• **Implementation**: AWS CloudFormation guard rule
• **Control behavior**: Proactive
• **Resource types**: AWS::RDS::DBCluster
• **AWS CloudFormation guard rule**: [CT.RDS.PR.25 rule specification](p. 1197)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.RDS.PR.25 rule specification](p. 1197)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.RDS.PR.25 example templates](p. 1200)

**Explanation**

Amazon RDS databases should have relevant logs enabled. Database logging provides detailed records of requests made to RDS. Database logs can assist with security and access audits, and they can help to diagnose availability issues.

**Usage considerations**

- This control applies only to Amazon RDS DB cluster engine types aurora, aurora-mysql, aurora-postgresql, mysql and postgres.

**Remediation for rule failure**

Specify EnableCloudwatchLogsExports with a list of all supported log types for the Amazon RDS database cluster engine.

The examples that follow show how to implement this remediation.

**Amazon RDS database (DB) Cluster - Example One**

Amazon RDS Aurora DB cluster configured with all available log types enabled for export to Amazon CloudWatch Logs. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "DBCluster": {
    "Type": "AWS::RDS::DBCluster",
    "Properties": {
      "Engine": "aurora",
      "MasterUsername": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBClusterSecret}:username}}"
      },
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBClusterSecret}:password}}"
      },
      "DBSubnetGroupName": {
        "Ref": "DBSubnetGroup"
      },
      "EnableCloudwatchLogsExports": [
        "audit",
        "error",
        "general",
        "slowquery"
      ]
  }
}
```

1195
YAML example

```yaml
DBCluster:
  Type: AWS::RDS::DBCluster
  Properties:
    Engine: aurora
    MasterUsername: !Sub '{{resolve:secretsmanager:${DBClusterSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${DBClusterSecret}::password}}'
    DBSubnetGroupName: !Ref 'DBSubnetGroup'
    EnableCloudwatchLogsExports:
      - audit
      - error
      - general
      - slowquery
```

The examples that follow show how to implement this remediation.

**Amazon RDS DB Cluster - Example Two**

Amazon RDS Multi-AZ Postgres DB cluster configured with all available log types enabled for export to Amazon CloudWatch Logs. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "DBCluster": {
    "Type": "AWS::RDS::DBCluster",
    "Properties": {
      "Engine": "aurora",
      "MasterUsername": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBClusterSecret}::username}}"
      },
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${DBClusterSecret}::password}}"
      },
      "DBSubnetGroupName": {
        "Ref": "DBSubnetGroup"
      },
      "EnableCloudwatchLogsExports": [
        "audit",
        "error",
        "general",
        "slowquery"
      ]
    }
  }
}
```

**YAML example**
CT.RDS.PR.25 rule specification

# Rule Specification

Rule Identifier:
rds_cluster_logging_enabled_check

Description:
This control checks whether Amazon RDS database clusters have all available log types enabled for export to Amazon CloudWatch Logs.

Reports on:
AWS::RDS::DBCluster

Evaluates:
AWS CloudFormation, AWS CloudFormation hook

Rule Parameters:
None

Scenarios:
1. Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
   And: The input document does not contain any RDS DB cluster resources
   Then: SKIP

2. Given: The input document contains an RDS DB cluster Resource
   And: 'Engine' is not one of 'aurora', 'aurora-mysql', 'aurora-postgresql', 'mysql' or 'postgres'
   Then: SKIP

3. Given: The input document contains an RDS DB cluster resource
   And: 'EnableCloudwatchLogsExports' has not been specified or has been specified as an empty list
   Then: FAIL

4. Given: The input document contains an RDS DB cluster resource
   And: The input document contains an RDS DB cluster resource
# And: 'Engine' is one of 'aurora', 'aurora-mysql', 'aurora-postgresql', 'mysql' or
# 'postgres':
# And: 'EnableCloudwatchLogsExports' has been specified and is a non-empty list
# And: One or more log types in 'EnableCloudwatchLogsExports' are not supported by
# the specified 'Engine'
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB cluster resource
# And: 'Engine' is one of 'aurora', 'aurora-mysql', 'aurora-postgresql', 'mysql' or
# 'postgres':
# And: 'EnableCloudwatchLogsExports' has been specified and is a non-empty list
# And: 'EnableCloudwatchLogsExports' does not contain all log types supported by the
# specified 'Engine'
# Then: FAIL
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB cluster resource
# And: 'Engine' is one of 'aurora', 'aurora-mysql' or 'mysql'
# And: 'EnableCloudwatchLogsExports' has been specified as a list with all supported
# log types
# for the 'Engine' ('audit', 'error', 'general' and 'slowquery')
# Then: PASS
# Scenario: 7
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB cluster resource
# And: 'Engine' is 'aurora-postgresql'
# And: 'EnableCloudwatchLogsExports' has been specified as a list with all supported
# log types
# for the 'Engine' ('postgresql')
# Then: PASS
# Scenario: 8
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an RDS DB cluster resource
# And: 'Engine' is 'postgres'
# And: 'EnableCloudwatchLogsExports' has been specified as a list with all supported
# log types
# for the 'Engine' ('postgresql', 'upgrade')
# Then: PASS

# Constants
#
let RDS_DB_CLUSTER_TYPE = "AWS::RDS::DBCluster"
let INPUT_DOCUMENT = this

let SUPPORTED_RDS_CLUSTER_ENGINES = [
  "aurora", "aurora-mysql", "aurora-postgresql", "mysql", "postgres"
]

let MYSQL_ENGINE_SUBTYPES = [ "aurora", "aurora-mysql", "mysql" ]
let AURORA_POSTGRES_ENGINE_SUBTYPES = [ "aurora-postgresql" ]
let POSTGRES_ENGINE_SUBTYPES = [ "postgres" ]

let MYSQL_SUPPORTED_LOG_TYPES = [ "audit", "error", "general", "slowquery" ]
let AURORA_POSTGRES_SUPPORTED_LOG_TYPES = [ "postgresql" ]
let POSTGRES_SUPPORTED_LOG_TYPES = [ "postgresql", "upgrade" ]

# Assignments
#
let rds_db_clusters = Resources.*[ Type == %RDS_DB_CLUSTER_TYPE ]
Proactive controls

# Primary Rules

rule rds_cluster_logging_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
    %rds_db_clusters not empty {
        check(%rds_db_clusters.Properties)
        [CT.RDS.PR.25]: Require an Amazon RDS database cluster to have logging configured
        [FIX]: Specify 'EnableCloudwatchLogsExports' with a list of all supported log
        types for the Amazon RDS database cluster engine.
        >>
    }

rule rds_cluster_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %RDS_DB_CLUSTER_TYPE) {
    check(%INPUT_DOCUMENT.%RDS_DB_CLUSTER_TYPE.resourceProperties)
    [CT.RDS.PR.25]: Require an Amazon RDS database cluster to have logging configured
    [FIX]: Specify 'EnableCloudwatchLogsExports' with a list of all supported log
    types for the Amazon RDS database cluster engine.
    >>
}

# Parameterized Rules

rule check(rds_db_cluster) {
    %rds_db_cluster[
        # Scenario 3
        filter_engine(this)
    ]{
        # Scenario 3
        EnableCloudwatchLogsExports exists
        check_is_list_and_not_empty(EnableCloudwatchLogsExports)
        # Scenario 4 and 6
        when Engine IN %MYSQL_ENGINE_SUBTYPES {
            %MYSQL_SUPPORTED_LOG_TYPES.* IN EnableCloudwatchLogsExports[*]
            EnableCloudwatchLogsExports.* IN %MYSQL_SUPPORTED_LOG_TYPES[*]
        }
        # Scenario 4 and 7
        when Engine IN %AURORA_POSTGRES_ENGINE_SUBTYPES {
            %AURORA_POSTGRES_SUPPORTED_LOG_TYPES.* IN EnableCloudwatchLogsExports[*]
            EnableCloudwatchLogsExports.* IN %AURORA_POSTGRES_SUPPORTED_LOG_TYPES[*]
        }
        # Scenario 4 and 8
        when Engine IN %POSTGRES_ENGINE_SUBTYPES {
            %POSTGRES_SUPPORTED_LOG_TYPES.* in EnableCloudwatchLogsExports[*]
            EnableCloudwatchLogsExports.* IN %POSTGRES_SUPPORTED_LOG_TYPES[*]
        }
    }
}

rule filter_engine(db_properties) {
    %db_properties[
        # Scenario 2
        Engine exists
        Engine in %SUPPORTED_RDS_CLUSTER_ENGINES
    ]
}

# Utility Rules

CT.RDS.PR.25 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
DBCluster:
  Type: AWS::RDS::DBCluster
  Properties:
    MasterUsername: exampleusername
    MasterUserPassword: example-password
    DBSubnetGroupName: example-db-subnet-group
    Engine: aurora
    EnableCloudwatchLogsExports:
      - audit
      - error
      - general
      - slowquery

PASS Example - Use this template to verify a compliant resource creation.

Resources:
DBCluster:
  Type: AWS::RDS::DBCluster
  Properties:
    DBClusterInstanceClass: db.m6gd.large
    MasterUsername: exampleusername
    MasterUserPassword: example-password
    DBSubnetGroupName: example-db-subnet-group
    Engine: postgres
    AllocatedStorage: 100
    StorageType: io1
    Iops: 3000
    EnableCloudwatchLogsExports:
      - postgresql
      - upgrade
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

<table>
<thead>
<tr>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCluster:</td>
</tr>
<tr>
<td>Type: AWS::RDS::DBCluster</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>MasterUsername: exampleusername</td>
</tr>
<tr>
<td>MasterUserPassword: example-password</td>
</tr>
<tr>
<td>DBSubnetGroupName: example-db-subnet-group</td>
</tr>
<tr>
<td>Engine: aurora</td>
</tr>
</tbody>
</table>

Amazon Redshift controls

Topics
- [CT.REDSHIFT.PR.1] Require an Amazon Redshift cluster to prohibit public access (p. 1201)
- [CT.REDSHIFT.PR.2] Require an Amazon Redshift cluster to have automatic snapshots configured (p. 1205)
- [CT.REDSHIFT.PR.3] Require an Amazon Redshift cluster to have audit logging configured (p. 1209)
- [CT.REDSHIFT.PR.4] Require an Amazon Redshift cluster to have automatic upgrades to major versions configured (p. 1214)
- [CT.REDSHIFT.PR.5] Require an Amazon Redshift cluster to have enhanced VPC routing (p. 1219)
- [CT.REDSHIFT.PR.6] Require an Amazon Redshift cluster to have a unique administrator username (p. 1223)
- [CT.REDSHIFT.PR.7] Require an Amazon Redshift cluster to have a unique database name (p. 1227)
- [CT.REDSHIFT.PR.8] Require an Amazon Redshift cluster to be encrypted (p. 1231)
- [CT.REDSHIFT.PR.9] Require that an Amazon Redshift cluster parameter group is configured to use Secure Sockets Layer (SSL) for encryption of data in transit (p. 1235)

[CT.REDSHIFT.PR.1] Require an Amazon Redshift cluster to prohibit public access

This control checks whether Amazon Redshift clusters are configured to prohibit public access.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Redshift::Cluster
- **AWS CloudFormation guard rule:** [CT.REDSHIFT.PR.1 rule specification (p. 1202)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.REDSHIFT.PR.1 rule specification (p. 1202)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.REDSHIFT.PR.1 example templates (p. 1204)]

Explanation

The PubliclyAccessible attribute of the Amazon Redshift cluster configuration indicates whether the cluster is publicly accessible. When the cluster is configured with the PubliclyAccessible parameter...
set to true, it is an internet-facing instance that has a publicly resolvable DNS name, which resolves to a public IP address.

When the cluster is not publicly accessible, it is an internal instance with a DNS name that resolves to a private IP address. Unless you intend for your cluster to be publicly accessible, the cluster should not be configured with PubliclyAccessible set to true.

**Remediation for rule failure**

Set PubliclyAccessible to false.

The examples that follow show how to implement this remediation.

**Amazon Redshift Cluster - Example**

Amazon Redshift cluster configured to prohibit public access. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "RedshiftCluster": {
        "Type": "AWS::Redshift::Cluster",
        "Properties": {
            "ClusterType": "single-node",
            "DBName": "sampledb",
            "MasterUsername": {
                "Fn::Sub": "{{resolve:secretsmanager:${RedshiftClusterSecret}::username}}"
            },
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager:${RedshiftClusterSecret}::password}}"
            },
            "NodeType": "ds2.xlarge",
            "PubliclyAccessible": false
        }
    }
}
```

**YAML example**

```yaml
RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: sampledb
    MasterUsername: !Sub '{{resolve:secretsmanager:${RedshiftClusterSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${RedshiftClusterSecret}::password}}'
    NodeType: ds2.xlarge
    PubliclyAccessible: false
```

**CT.REDSHIFT.PR.1 rule specification**

```bash
# #####################################################################
##       Rule Specification       ##
# #####################################################################
#
```
# Rule Identifier:
#   redshift_cluster_public_access_check
#
# Description:
#   This control checks whether Amazon Redshift clusters are configured to prohibit public access.
#
# Reports on:
#   AWS::Redshift::Cluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook document
#     And: The input document does not contain any Redshift cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook document
#     And: The input document contains a Redshift cluster resource
#     And: 'PubliclyAccessible' has not been specified
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook document
#     And: The input document contains a Redshift cluster resource
#     And: 'PubliclyAccessible' has been specified
#     And: 'PubliclyAccessible' has been set to bool(true)
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation document or AWS CloudFormation hook document
#     And: The input document contains a Redshift cluster resource
#     And: 'PubliclyAccessible' has been specified
#     And: 'PubliclyAccessible' has been set to bool(false)
#     Then: PASS
#
# Constants
#
let REDSHIFT_CLUSTER_TYPE = "AWS::Redshift::Cluster"
let INPUT_DOCUMENT = this
#
# Assignments
#
let redshift_clusters = Resources.*[ Type == %REDSHIFT_CLUSTER_TYPE ]
#
# Primary Rules
#
rule redshift_cluster_public_access_check when is_cfn_template(%INPUT_DOCUMENT)
  %redshift_clusters not empty {
    check(%redshift_clusters.Properties)
    <<
      [CT.REDSHIFT.PR.1]: Require an Amazon Redshift cluster to prohibit public access
      [FIX]: Set 'PubliclyAccessible' to 'false'.
    >>
  }
}
rule redshift_cluster_public_access_check when is_cfn_hook(%INPUT_DOCUMENT, %REDSHIFT_CLUSTER_TYPE) {
    check(%INPUT_DOCUMENT.%REDSHIFT_CLUSTER_TYPE.resourceProperties)
    <<
    [CT.REDSHIFT.PR.1]: Require an Amazon Redshift cluster to prohibit public access
    [FIX]: Set 'PubliclyAccessible' to 'false'.
    >>
}

# Parameterized Rules
#
rule check(redshift_cluster) {
    %redshift_cluster {
        # Scenario 2
        PubliclyAccessible exists
        # Scenario 3 and 4
        PubliclyAccessible == false
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.REDSHIFT.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
RedshiftSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: Redshift cluster secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "examplemasterusername"}'
      GenerateStringKey: password
      PasswordLength: 16
      ExcludeCharacters: ""@/\\"
RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: exampledb
    MasterUsername:
      Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
    MasterUserPassword:
      Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
    NodeType: dc2.large
    PubliclyAccessible: false
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:

RedshiftSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: Redshift cluster secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "examplemasterusername"}'
      GenerateStringKey: password
      PasswordLength: 16
      ExcludeCharacters: "'"@/\"

RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: exampledb
    MasterUsername: Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
    MasterUserPassword: Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
    NodeType: dc2.large
    PubliclyAccessible: true

[CT.REDSHIFT.PR.2] Require an Amazon Redshift cluster to have automatic snapshots configured

This control checks whether Amazon Redshift clusters have automated snapshots enabled, and that the clusters are set with an automated snapshot retention period greater than or equal to seven (7) days.

- **Control objective:** Improve resiliency
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Redshift::Cluster
- **AWS CloudFormation guard rule:** CT.REDSHIFT.PR.2 rule specification (p. 1206)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.REDSHIFT.PR.2 rule specification (p. 1206)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.REDSHIFT.PR.2 example templates (p. 1208)

Explanation

Backups help you to recover more quickly from a security incident. They strengthen the resilience of your systems. Amazon Redshift takes periodic snapshots by default. This control checks whether automatic snapshots are created and retained for at least seven days.

**Remediation for rule failure**

Set AutomatedSnapshotRetentionPeriod to an integer value greater than or equal to 7 days.
The examples that follow show how to implement this remediation.

Amazon Redshift Cluster - Example

Amazon Redshift cluster configured with automatic snapshots active. The example is shown in JSON and in YAML.

JSON example

```
{
    "RedshiftCluster": {
        "Type": "AWS::Redshift::Cluster",
        "Properties": {
            "ClusterType": "single-node",
            "DBName": "sampledb",
            "MasterUsername": {
                "Fn::Sub": "{{resolve:secretsmanager:${RedshiftSecret}::username}}"
            },
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager:${RedshiftSecret}::password}}"
            },
            "NodeType": "ds2.xlarge",
            "AutomatedSnapshotRetentionPeriod": 7
        }
    }
}
```

YAML example

```
RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: sampledb
    MasterUsername: !Sub "{{resolve:secretsmanager:${RedshiftSecret}::username}}"
    MasterUserPassword: !Sub "{{resolve:secretsmanager:${RedshiftSecret}::password}}"
    NodeType: ds2.xlarge
    AutomatedSnapshotRetentionPeriod: 7
```

CT.REDSHIFT.PR.2 rule specification

```
# ###################################
##       Rule Specification        ##
####################################
#
# Rule Identifier:  
#   redshift_backup_enabled_check 
#
# Description:  
#   This control checks whether Amazon Redshift clusters have automated snapshots enabled, and that the clusters are set with an automated snapshot retention period greater than or equal to seven (7) days. 
#
# Reports on:  
#   AWS::Redshift::Cluster
```

1206
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document does not contain any Redshift cluster resources
#   Then: SKIP
# Scenario: 2
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a Redshift cluster resource
#   And: 'AutomatedSnapshotRetentionPeriod' has not been specified
#   Then: FAIL
# Scenario: 3
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a Redshift cluster resource
#   And: 'AutomatedSnapshotRetentionPeriod' has been specified
#   And: 'AutomatedSnapshotRetentionPeriod' has been set to '0'
#   Then: FAIL
# Scenario: 4
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a Redshift cluster resource
#   And: 'AutomatedSnapshotRetentionPeriod' has been specified
#   And: 'AutomatedSnapshotRetentionPeriod' has been set to a value <7
#   Then: FAIL
# Scenario: 5
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a Redshift cluster resource
#   And: 'AutomatedSnapshotRetentionPeriod' has been specified
#   And: 'AutomatedSnapshotRetentionPeriod' has been set to a value >= 7
#   Then: PASS

# Constants
#
let REDSHIFT_CLUSTER_TYPE = "AWS::Redshift::Cluster"
let INPUT_DOCUMENT = this

# Assignments
#
let redshift_clusters = Resources.*[ Type == %REDSHIFT_CLUSTER_TYPE ]

# Primary Rules
#
rule redshift_backup_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
  %redshift_clusters not empty {
    check(%redshift_clusters.Properties)
    <<
      [CT.REDSHIFT.PR.2]: Require an Amazon Redshift cluster to have automatic snapshots configured
      [FIX]: Set 'AutomatedSnapshotRetentionPeriod' to an integer value greater than or equal to 7 days.
    >>
  }
AWS Control Tower User Guide
Proactive controls

```plaintext
rule redshift_backup_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %REDSHIFT_CLUSTER_TYPE) {
    check(%INPUT_DOCUMENT.%REDSHIFT_CLUSTER_TYPE.resourceProperties)
    <<
    [CT.REDSHIFT.PR.2]: Require an Amazon Redshift cluster to have automatic snapshots configured
    [FIX]: Set 'AutomatedSnapshotRetentionPeriod' to an integer value greater than or equal to 7 days.
    >>
}
#
# Parameterized Rules
#
rule check(redshift_cluster) {
    %redshift_cluster {
        # Scenario 2
        AutomatedSnapshotRetentionPeriod exists
        # Scenario 3, 4 and 5
        AutomatedSnapshotRetentionPeriod >= 7
    }
}
#
# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

**CT.REDSHIFT.PR.2 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```
Resources:
  RedshiftSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: Redshift cluster secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "examplemasterusername"}'
        GenerateStringKey: password
        PasswordLength: 16
        ExcludeCharacters: "'"@\/
  RedshiftCluster:
    Type: AWS::Redshift::Cluster
    Properties:
      ClusterType: single-node
      DBName: exampledb
      MasterUsername: 
        Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
      MasterUserPassword:
```

1208
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
RedshiftSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: Redshift cluster secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "examplemasterusername"}'
      GenerateStringKey: password
      PasswordLength: 16
      ExcludeCharacters: "'"@/\\"

RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: exampledb
    MasterUsername:
      Fn::Sub: '{\{resolve:secretsmanager:\${RedshiftSecret}::username\\}}'
    MasterUserPassword:
      Fn::Sub: '{\{resolve:secretsmanager:\${RedshiftSecret}::password\\}}'
    NodeType: dc2.large
    PubliclyAccessible: false
    AutomatedSnapshotRetentionPeriod: 5

[CT.REDSHIFT.PR.3] Require an Amazon Redshift cluster to have audit logging configured

This control checks whether an Amazon Redshift cluster has audit logging activated.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Redshift::Cluster
- **AWS CloudFormation guard rule:** [CT.REDSHIFT.PR.3 rule specification](p. 1210)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.REDSHIFT.PR.3 rule specification](p. 1210)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.REDSHIFT.PR.3 example templates](p. 1213)

Explanation

Amazon Redshift audit logging provides additional information about connections and user activities in your cluster. This data can be stored and secured in Amazon S3, and it can be helpful for security audits and investigations.
Remediation for rule failure

Provide a `LoggingProperties` configuration and set `BucketName` to the name of an Amazon S3 bucket configured to receive Amazon Redshift audit logs.

The examples that follow show how to implement this remediation.

Amazon Redshift Cluster - Example

Amazon Redshift cluster configured with audit logging enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "RedshiftCluster": {
      "Type": "AWS::Redshift::Cluster",
      "Properties": {
         "ClusterType": "single-node",
         "DBName": "sampledb",
         "MasterUsername": {
            "Fn::Sub": "{{resolve:secretsmanager:${RedshiftSecret}::username}}"
         },
         "MasterUserPassword": {
            "Fn::Sub": "{{resolve:secretsmanager:${RedshiftSecret}::password}}"
         },
         "NodeType": "dc2.large",
         "PubliclyAccessible": false,
         "LoggingProperties": {
            "BucketName": {
               "Ref": "S3Bucket"
            },
            "S3KeyPrefix": "sample-cluster-logs"
         }
      }
   }
}
```

**YAML example**

```yaml
RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: sampledb
    MasterUsername: !Sub '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
    NodeType: dc2.large
    PubliclyAccessible: false
    LoggingProperties:
      BucketName: !Ref 'S3Bucket'
      S3KeyPrefix: sample-cluster-logs
```

**CT.REDSHIFT.PR.3 rule specification**

```
# ............................................................
##       Rule Specification       ##
```

1210
# Rule Identifier:
# redshift_cluster_audit_logging_enabled_check
#
# Description:
# This control checks whether an Amazon Redshift cluster has audit logging activated.
#
# Reports on:
# AWS::Redshift::Cluster
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document does not contain any Redshift cluster resources
#   Then: SKIP
# Scenario: 2
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a Redshift cluster resource
#   And: 'LoggingProperties' has not been specified
#   Then: FAIL
# Scenario: 3
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a Redshift cluster resource
#   And: 'LoggingProperties' has been specified
#   And: 'BucketName' on 'LoggingProperties' has been specified and is an empty string
#   or invalid local reference
#   Then: FAIL
# Scenario: 4
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#   And: The input document contains a Redshift cluster resource
#   And: 'LoggingProperties' has been specified
#   And: 'BucketName' on 'LoggingProperties' has been specified and is a non-empty
#   string or valid local reference
#   Then: PASS
#
# Constants
#
let REDSHIFT_CLUSTER_TYPE = "AWS::Redshift::Cluster"
let INPUT_DOCUMENT = this
#
# Assignments
#
let redshift_clusters = Resources.*[ Type == %REDSHIFT_CLUSTER_TYPE ]
#
# Primary Rules
#
rule redshift_cluster_audit_logging_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%redshift_clusters not empty {
  check(%redshift_clusters.Properties)
  <<
  [CT.REDSHIFT.PR.3]: Require an Amazon Redshift cluster to have audit logging
configured
}
[FIX]: Provide a 'LoggingProperties' configuration and set 'BucketName' to the name of an Amazon S3 bucket configured to receive Amazon Redshift audit logs.

```java
rule redshift_cluster_audit_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %REDSHIFT_CLUSTER_TYPE) {
    check(%INPUT_DOCUMENT.%REDSHIFT_CLUSTER_TYPE.resourceProperties)

    [CT.REDSHIFT.PR.3]: Require an Amazon Redshift cluster to have audit logging configured
    [FIX]: Provide a 'LoggingProperties' configuration and set 'BucketName' to the name of an Amazon S3 bucket configured to receive Amazon Redshift audit logs.
}
```

# Parameterized Rules

```java
rule check(redshift_cluster) {
    %redshift_cluster {
        # Scenario 2
        LoggingProperties exists
        LoggingProperties is_struct
        LoggingProperties {
            # Scenario 3 and 4
            BucketName exists
            check_is_string_and_not_empty(BucketName) or
            check_local_references(%INPUT_DOCUMENT, BucketName, "AWS::S3::Bucket")
        }
    }
}
```

# Utility Rules

```java
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

rule check_is_string_and_not_empty(value) {
    %value {
        this is_string
        this != /\A\s*\z/
    }
}

rule check_local_references(doc, reference_properties, referenced_resource_type) {
    %reference_properties {
        'Fn::GetAtt' {
            query_for_resource(%doc, this[0], %referenced_resource_type)
            <<Local Stack reference was invalid>>
        } or Ref {
            query_for_resource(%doc, this, %referenced_resource_type)
            <<Local Stack reference was invalid>>
        }
    }
}
CT.REDSHIFT.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
S3Bucket:
  Type: AWS::S3::Bucket
BucketPolicy:
  Type: AWS::S3::BucketPolicy
Properties:
  Bucket:
    Ref: S3Bucket
  PolicyDocument:
    Version: 2012-10-17
    Statement:
      - Effect: Allow
        Principal:
          Service: redshift.amazonaws.com
        Action:
          - s3:GetBucketAcl
        Resource:
          - Fn::GetAtt: - S3Bucket - Arn
      - Effect: Allow
        Principal:
          Service: redshift.amazonaws.com
        Action:
          - s3:PutObject
        Resource:
          - Fn::Join: -."
          - - Fn::GetAtt: - S3Bucket - Arn
        -/*

RedshiftSecret:
  Type: AWS::SecretsManager::Secret
Properties:
  Description: Redshift cluster secret
  GenerateSecretString:
    SecretStringTemplate: '{"username": "examplemasterusername"}'
    GenerateStringKey: password
    PasswordLength: 16
    ExcludeCharacters: "'"@/\"

RedshiftCluster:
  Type: AWS::Redshift::Cluster
Properties:
  ClusterType: single-node
  DBName: exampledb
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
RedshiftSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
  - Description: Redshift cluster secret
  - GenerateSecretString:
    - SecretStringTemplate: '{"username": "examplemasterusername"}'
    - GenerateStringKey: password
    - PasswordLength: 16
    - ExcludeCharacters: '"@/\"'
RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
  - ClusterType: single-node
  - DBName: exampledb
  - MasterUsername:
    - Fn::Sub: '{resolve:secretsmanager:${RedshiftSecret}::username}'
  - MasterUserPassword:
    - Fn::Sub: '{resolve:secretsmanager:${RedshiftSecret}::password}'
  - NodeType: dc2.large
  - PubliclyAccessible: false

[CT.REDSHIFT.PR.4] Require an Amazon Redshift cluster to have automatic upgrades to major versions configured

This control checks whether automatic major version upgrades are enabled for your Amazon Redshift cluster.

- **Control objective:** Manage vulnerabilities
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Redshift::Cluster
- **AWS CloudFormation guard rule:** [CT.REDSHIFT.PR.4 rule specification (p. 1216)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.REDSHIFT.PR.4 rule specification (p. 1216)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.REDSHIFT.PR.4 example templates (p. 1218)]
Explanation

Enabling automatic major version upgrades ensures that the latest major version updates to Amazon Redshift clusters are installed during the maintenance window. These updates might include security patches and bug fixes. Keeping up to date with patch installation is an important step in securing systems.

Remediation for rule failure

Set the AllowVersionUpgrade property to true or do not specify it (default).

The examples that follow show how to implement this remediation.

Amazon Redshift Cluster - Example One

Amazon Redshift cluster with automatic major version upgrades enabled through AWS CloudFormation defaults. The example is shown in JSON and in YAML.

JSON example

```json
{
  "RedshiftCluster": {
    "Type": "AWS::Redshift::Cluster",
    "Properties": {
      "ClusterType": "single-node",
      "DBName": "exampledb",
      "MasterUsername": {
        "Fn::Sub": "{{resolve:secretsmanager:${RedshiftSecret}::username}}"
      },
      "MasterUserPassword": {
        "Fn::Sub": "{{resolve:secretsmanager:${RedshiftSecret}::password}}"
      },
      "NodeType": "ds2.xlarge"
    }
  }
}
```

YAML example

```yaml
RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: exampledb
    MasterUsername: !Sub '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
    NodeType: ds2.xlarge
```

The examples that follow show how to implement this remediation.

Amazon Redshift Cluster - Example Two

Amazon Redshift cluster configured with automatic major version upgrades enabled through the AllowVersionUpgrade property. The example is shown in JSON and in YAML.

JSON example
YAML example

RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: exampledb
    MasterUsername: !Sub '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
    NodeType: ds2.xlarge
    AllowVersionUpgrade: true

CT.REDSHIFT.PR.4 rule specification

# ###################################################################
#               Rule Specification                                    #
# ###################################################################
#
# Rule Identifier:
#    redshift_cluster_allow_version_upgrade_check
#
# Description:
#    Checks whether automatic major version upgrades are enabled for the Amazon Redshift cluster.
#
# Reports on:
#    AWS::Redshift::Cluster
#
# Evaluates:
#    AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#    None
#
# Scenarios:
#    Scenario: 1
#      Given: The input document is an AWS CloudFormation or CloudFormation hook document
#      And: The input document does not contain any Redshift cluster resources
#      Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains a Redshift cluster resource
# And: 'AllowVersionUpgrade' has been provided
# And: 'AllowVersionUpgrade' has been set to bool(false)
# Then: FAIL

# Scenario: 3
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains a Redshift cluster resource
# And: 'AllowVersionUpgrade' has not been provided
# Then: PASS

# Scenario: 4
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains a Redshift cluster resource
# And: 'AllowVersionUpgrade' has been provided
# And: 'AllowVersionUpgrade' has been set to bool(true)
# Then: PASS

# Constants
#
let REDSHIFT_CLUSTER_TYPE = "AWS::Redshift::Cluster"
let INPUT_DOCUMENT = this
#
# Assignments
#
let redshift_clusters = Resources.*[ Type == %REDSHIFT_CLUSTER_TYPE ]
#
# Primary Rules
#
rule redshift_cluster_allow_version_upgrade_check when is_cfn_template(%INPUT_DOCUMENT)
%redshift_clusters not empty { 
  check(%redshift_clusters.Properties) <<
  [CT.REDSHIFT.PR.4]: Require an Amazon Redshift cluster to have automatic upgrades
to major versions configured
  [FIX]: Set the 'AllowVersionUpgrade' property to true or do not specify it (default).
  >>
}
rule redshift_cluster_allow_version_upgrade_check when is_cfn_hook(%INPUT_DOCUMENT,
%REDSHIFT_CLUSTER_TYPE) {
  check(%INPUT_DOCUMENT.%REDSHIFT_CLUSTER_TYPE.resourceProperties) <<
  [CT.REDSHIFT.PR.4]: Require an Amazon Redshift cluster to have automatic upgrades
to major versions configured
  [FIX]: Set the 'AllowVersionUpgrade' property to true or do not specify it (default).
  >>
}
#
# Parameterized Rules
#
rule check(redshift_cluster) {
  %redshift_cluster {
  # Scenario 3
  AllowVersionUpgrade not exists or
  # Scenario 2 and 4
  AllowVersionUpgrade == true
  }
}
# Utility Rules

## Proactive controls

### CT.REDSHIFT.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```
Resources:
  RedshiftSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: Redshift cluster secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "examplemasterusername"}'
        GenerateStringKey: password
        PasswordLength: 16
        ExcludeCharacters: "'"@/\"

  RedshiftCluster:
    Type: AWS::Redshift::Cluster
    Properties:
      ClusterType: single-node
      DBName: exampledb
      MasterUsername:
        Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
      MasterUserPassword:
        Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
      NodeType: dc2.large
      PubliclyAccessible: false
```

**FAIL Example** - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
  RedshiftSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: Redshift cluster secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "examplemasterusername"}'
        GenerateStringKey: password
        PasswordLength: 16
        ExcludeCharacters: "'"@/\"

  RedshiftCluster:
    Type: AWS::Redshift::Cluster
    Properties:
```

1218
ClusterType: single-node
DBName: exampledb
MasterUsername:
  Fn::Sub: '{\{resolve:secretsmanager:${RedshiftSecret}::username\}}'
MasterUserPassword:
  Fn::Sub: '{\{resolve:secretsmanager:${RedshiftSecret}::password\}}'
NodeType: dc2.large
PubliclyAccessible: false
AllowVersionUpgrade: false

[CT.REDSHIFT.PR.5] Require an Amazon Redshift cluster to have enhanced VPC routing

This control checks whether an Amazon Redshift cluster has enhanced VPC routing configured.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Redshift::Cluster
- **AWS CloudFormation guard rule:** [CT.REDSHIFT.PR.5 rule specification (p. 1220)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.REDSHIFT.PR.5 rule specification (p. 1220)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.REDSHIFT.PR.5 example templates (p. 1222)]

**Explanation**

Enhanced VPC routing forces all copy and unload traffic between the cluster and the data repositories to go through your VPC. With enhanced routing active, you can use VPC features, such as security groups and network access control lists, to secure network traffic. You can also use VPC Flow Logs to monitor network traffic.

**Remediation for rule failure**

Set EnhancedVpcRouting to true.

The examples that follow show how to implement this remediation.

**Amazon Redshift Cluster - Example**

Amazon Redshift cluster configured with enhanced VPC routing. The example is shown in JSON and in YAML.

**JSON example**

```
{
    "RedshiftCluster": {
        "Type": "AWS::Redshift::Cluster",
        "Properties": {
            "ClusterType": "single-node",
            "DBName": "sampledb",
            
```
"MasterUsername": {
    "Fn::Sub": "{{resolve:secretsmanager:${RedshiftSecret}::username}}"
},
"MasterUserPassword": {
    "Fn::Sub": "{{resolve:secretsmanager:${RedshiftSecret}::password}}"
},
"NodeType": "ds2.xlarge",
"EnhancedVpcRouting": true
}
}

YAML example

RedshiftCluster:
    Type: AWS::Redshift::Cluster
    Properties:
        ClusterType: single-node
        DBName: sampledb
        MasterUsername: !Sub '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
        MasterUserPassword: !Sub '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
        NodeType: ds2.xlarge
        EnhancedVpcRouting: true

CT.REDSHIFT.PR.5 rule specification

# ###############################################################################
##       Rule Specification        ##
# ###############################################################################

# Rule Identifier:
#   redshift_enhanced_vpc_routing_enabled_check
#
# Description:
#   This control checks whether an Amazon Redshift cluster has enhanced VPC routing configured.
#
# Reports on:
#   AWS::Redshift::Cluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any Redshift cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a Redshift cluster resource
#     And: 'EnhancedVpcRouting' has not been specified
#     Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a Redshift cluster resource
# And: 'EnhancedVpcRouting' has been specified
# And: 'EnhancedVpcRouting' has been set to bool(false)
# Then: FAIL

# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a Redshift cluster resource
# And: 'EnhancedVpcRouting' has been specified
# And: 'EnhancedVpcRouting' has been set to bool(true)
# Then: PASS

# Constants

let REDSHIFT_CLUSTER_TYPE = "AWS::Redshift::Cluster"
let INPUT_DOCUMENT = this

# Assignments

let redshift_clusters = Resources.*[ Type == %REDSHIFT_CLUSTER_TYPE ]

# Primary Rules

rule redshift_enhanced_vpc_routing_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
%redshift_clusters not empty {
  check(%redshift_clusters.Properties)
  %redshift_clusters not empty {
    =>
    [CT.REDSHIFT.PR.5]: Require an Amazon Redshift cluster to have enhanced VPC routing
    [FIX]: Set 'EnhancedVpcRouting' to 'true'.
    =>
  }
}

rule redshift_enhanced_vpc_routing_enabled_check when is_cfn_hook(%INPUT_DOCUMENT,
%REDSHIFT_CLUSTER_TYPE) {
  check(%INPUT_DOCUMENT.%REDSHIFT_CLUSTER_TYPE.resourceProperties)
  =>
  [CT.REDSHIFT.PR.5]: Require an Amazon Redshift cluster to have enhanced VPC routing
  [FIX]: Set 'EnhancedVpcRouting' to 'true'.
  =>
}

# Parameterized Rules

rule check(redshift_cluster) {
  %redshift_cluster {
    # Scenario 2
    EnhancedVpcRouting exists
    # Scenario 3 and 4
    EnhancedVpcRouting == true
  }
}

# Utility Rules

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }

#### CT.REDSHIFT.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```yaml
Resources:
  RedshiftSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: Redshift cluster secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "examplemasterusername"}
        GenerateStringKey: password
        PasswordLength: 16
        ExcludeCharacters: "@/\"
  RedshiftCluster:
    Type: AWS::Redshift::Cluster
    Properties:
      ClusterType: single-node
      DBName: exampledb
      MasterUsername:
        Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
      MasterUserPassword:
        Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
      NodeType: dc2.large
      PubliclyAccessible: false
      EnhancedVpcRouting: true
```

**FAIL Example** - Use this template to verify that the control prevents non-compliant resource creation.

```yaml
Resources:
  RedshiftSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Description: Redshift cluster secret
      GenerateSecretString:
        SecretStringTemplate: '{"username": "examplemasterusername"}
        GenerateStringKey: password
        PasswordLength: 16
        ExcludeCharacters: "@/\"
  RedshiftCluster:
    Type: AWS::Redshift::Cluster
    Properties:
      ClusterType: single-node
      DBName: exampledb
      MasterUsername:
        Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
      MasterUserPassword:
        Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
```

1222
[CT.REDSHIFT.PR.6] Require an Amazon Redshift cluster to have a unique administrator username

This control checks whether an Amazon Redshift cluster has changed the administrator username from its default value.

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Redshift::Cluster
- **AWS CloudFormation guard rule:** [CT.REDSHIFT.PR.6 rule specification](p. 1224)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.REDSHIFT.PR.6 rule specification](p. 1224)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.REDSHIFT.PR.6 example templates](p. 1226)

**Explanation**

When creating an Amazon Redshift cluster, you should change the default administrator username to a unique value. Default usernames are public knowledge, so they should be changed upon configuration. Changing the default username reduces the risk of unintended access.

**Remediation for rule failure**

Set **MasterUsername** to a value other than **awsuser**.

The examples that follow show how to implement this remediation.

**Amazon Redshift Cluster - Example**

Amazon Redshift cluster configured with an administrator username different from the default value of **awsuser**. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "RedshiftCluster": {
        "Type": "AWS::Redshift::Cluster",
        "Properties": {
            "ClusterType": "single-node",
            "DBName": "sampledb",
            "MasterUsername": "samplemasterusername",
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager:${RedshiftClusterSecret}::password}}",
            },
            "NodeType": "ds2.xlarge"
        }
    }
}
```
YAML example

RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: sampledb
    MasterUsername: samplemasterusername
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${RedshiftClusterSecret}::password}}'
    NodeType: ds2.xlarge

CT.REDSHIFT.PR.6 rule specification

```
# ####################################################################
##       Rule Specification        ##
# ####################################################################
#
# Rule Identifier:
#   redshift_default_admin_check
#
# Description:
#   This control checks whether an Amazon Redshift cluster has changed the administrator
#   username from its default value.
#
# Reports on:
#   AWS::Redshift::Cluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#            And: The input document does not contain any Redshift cluster resources
#            Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#            And: The input document contains a Redshift cluster resource
#            And: 'MasterUsername' has not been specified
#            Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#            And: The input document contains a Redshift cluster resource
#            And: 'MasterUsername' has been specified
#            And: 'MasterUsername' has been set to 'awsuser'
#            Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
```
And: The input document contains a Redshift cluster resource
And: 'MasterUsername' has been specified
And: 'MasterUsername' has been set to a value not equal to 'awsuser'
Then: PASS

# Constants
#
let REDSHIFT_CLUSTER_TYPE = "AWS::Redshift::Cluster"
let INPUT_DOCUMENT = this
#
# Assignments
#
let redshift_clusters = Resources.*[ Type == %REDSHIFT_CLUSTER_TYPE ]
#
# Primary Rules
#
rule redshift_default_admin_check when is_cfn_template(%INPUT_DOCUMENT)
%redshift_clusters not empty {
  check(%redshift_clusters.Properties)
  <<
    [CT.REDSHIFT.PR.6]: Require an Amazon Redshift cluster to have a unique administrator username
    [FIX]: Set 'MasterUsername' to a value other than 'awsuser'.
  >>
}
rule redshift_default_admin_check when is_cfn_hook(%INPUT_DOCUMENT, %REDSHIFT_CLUSTER_TYPE)
{
  check(%INPUT_DOCUMENT.%REDSHIFT_CLUSTER_TYPE.resourceProperties)
  <<
    [CT.REDSHIFT.PR.6]: Require an Amazon Redshift cluster to have a unique administrator username
    [FIX]: Set 'MasterUsername' to a value other than 'awsuser'.
  >>
}
#
# Parameterized Rules
#
rule check(redshift_cluster) {
  %redshift_cluster {
    # Scenario 2
    MasterUsername exists
    check_is_string_and_not_empty(MasterUsername)
    # Scenario 3 and 4
    MasterUsername != "awsuser"
  }
}
#
# Utility Rules
#
rule check_is_string_and_not_empty(value) {
  %value {
    this is_string
    this !/^\A\s*\z/
  }
}
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
RedshiftSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: Redshift cluster secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "examplemasterusername"}'
      GenerateStringKey: password
      PasswordLength: 16
      ExcludeCharacters: '"@/\'
RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: exampledb
    MasterUsername: examplemasterusername
    MasterUserPassword:
      Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
    NodeType: dc2.large
    PubliclyAccessible: false

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
RedshiftSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: Redshift cluster secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "awsuser"}'
      GenerateStringKey: password
      PasswordLength: 16
      ExcludeCharacters: '"@/\'
RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: exampledb
    MasterUsername: awsuser
    MasterUserPassword:
      Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
    NodeType: dc2.large
    PubliclyAccessible: false
[CT.REDSHIFT.PR.7] Require an Amazon Redshift cluster to have a unique database name

This control checks whether an Amazon Redshift cluster has changed its database name from the default value.

- **Control objective:** Protect configurations
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Redshift::Cluster
- **AWS CloudFormation guard rule:** [CT.REDSHIFT.PR.7 rule specification (p. 1228)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.REDSHIFT.PR.7 rule specification (p. 1228)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.REDSHIFT.PR.7 example templates (p. 1230)]

**Explanation**

When creating a Redshift cluster, you should change the default database name to a unique value. Default names are public knowledge, so they should be changed upon configuration. For example, a well-known name can lead to inadvertent access, if included in IAM policy conditions.

**Remediation for rule failure**

Set `DBName` to a database name that is different from the default value of `dev`.

The examples that follow show how to implement this remediation.

**Amazon Redshift Cluster - Example**

Amazon Redshift cluster configured with a database name different from the default value of `dev`. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "RedshiftCluster": {
        "Type": "AWS::Redshift::Cluster",
        "Properties": {
            "ClusterType": "single-node",
            "DBName": "sampledb",
            "MasterUsername": {
                "Fn::Sub": "{{resolve:secretsmanager:${RedshiftClusterSecret}\::username}}"
            },
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager:${RedshiftClusterSecret}\::password}}"
            },
            "NodeType": "dc2.large",
            "PubliclyAccessible": false
        }
    }
}
```
YAML example

RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: sampledb
    MasterUsername: !Sub '{{resolve:secretsmanager:${RedshiftClusterSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${RedshiftClusterSecret}::password}}'
    NodeType: dc2.large
    PubliclyAccessible: false

CT.REDSHIFT.PR.7 rule specification

# ###################################################################
##       Rule Specification        ##
# ###################################################################
#
# Rule Identifier:
#   redshift_default_db_name_check
#
# Description:
#   This control checks whether an Amazon Redshift cluster has changed its database name from the default value.
#
# Reports on:
#   AWS::Redshift::Cluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any Redshift cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains a Redshift cluster resource
#     And: The 'DBName' property has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains Redshift cluster resource
#     And: The 'DBName' property has been provided with a value of 'dev' or an empty string
#     Then: FAIL
#   Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains Redshift cluster resource
# And: The 'DBName' property has been provided with a non-empty string that is not
equal to 'dev'
# Then: PASS

# Constants

let RESOURCE_TYPE = "AWS::Redshift::Cluster"
let INPUT_DOCUMENT = this
let INVALID_DB_NAME_STRING = "dev"

# Assignments

let redshift_clusters = Resources.*[ Type == %RESOURCE_TYPE ]

# Primary Rules

rule redshift_default_db_name_check when is_cfn_template(%INPUT_DOCUMENT)
%redshift_clusters not empty {
  check_db_name(%redshift_clusters.Properties)
  <<
  [CT.REDSHIFT.PR.7]: Require an Amazon Redshift cluster to have a unique database
  name
  [FIX]: Set 'DBName' to a database name that is different from the default value of
  'dev'.
  >>
}

rule redshift_default_db_name_check when is_cfn_hook(%INPUT_DOCUMENT, %RESOURCE_TYPE) {
  check_db_name(%INPUT_DOCUMENT.%RESOURCE_TYPE.resourceProperties)
  <<
  [CT.REDSHIFT.PR.7]: Require an Amazon Redshift cluster to have a unique database
  name
  [FIX]: Set 'DBName' to a database name that is different from the default value of
  'dev'.
  >>
}

# Parameterized Rules

rule check_db_name(redshift_cluster) {
  %redshift_cluster {
    # Scenario 2
    DBName exists

    # Scenario 3 and 4
    check_is_string_and_not_empty(DBName)
    DBName != %INVALID_DB_NAME_STRING
  }
}

# Utility Rules

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
CT.REDSHIFT.PR.7 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
- **RedshiftSecret**:
  - Type: AWS::SecretsManager::Secret
  - Properties:
    - Description: Redshift cluster secret
    - GenerateSecretString:
      - SecretStringTemplate: '{"username": "examplemasterusername"}'
      - GenerateStringKey: password
      - PasswordLength: 16
      - ExcludeCharacters: "'"@/\"

- **RedshiftCluster**:
  - Type: AWS::Redshift::Cluster
  - Properties:
    - ClusterType: "single-node"
    - MasterUsername:
      - Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
    - MasterUserPassword:
      - Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
    - NodeType: dc2.large
    - PubliclyAccessible: false
    - DBName: exampledb

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
- **RedshiftSecret**:
  - Type: AWS::SecretsManager::Secret
  - Properties:
    - Description: Redshift cluster secret
    - GenerateSecretString:
      - SecretStringTemplate: '{"username": "examplemasterusername"}'
      - GenerateStringKey: password
      - PasswordLength: 16
      - ExcludeCharacters: "'"@/\"

- **RedshiftCluster**:
  - Type: AWS::Redshift::Cluster
  - Properties:
    - ClusterType: "single-node"
    - MasterUsername:
      - Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
    - MasterUserPassword:
      - Fn::Sub: '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
    - NodeType: dc2.large
    - PubliclyAccessible: false
    - DBName: exampledb
[CT.REDSHIFT.PR.8] Require an Amazon Redshift cluster to be encrypted

This control checks whether an Amazon Redshift cluster is encrypted.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Redshift::Cluster
- **AWS CloudFormation guard rule:** CT.REDSHIFT.PR.8 rule specification (p. 1232)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.REDSHIFT.PR.8 rule specification (p. 1232)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.REDSHIFT.PR.8 example templates (p. 1234)

Explanation

In Amazon Redshift, you can enable database encryption for your clusters, which helps protect data at rest. When you enable encryption for a cluster, the data blocks and system metadata are encrypted for the cluster and its snapshots.

**Remediation for rule failure**

Set the value of the Encrypted property to true.

The examples that follow show how to implement this remediation.

Amazon Redshift Cluster - Example

An Amazon Redshift cluster configured with encryption enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "RedshiftCluster": {
        "Type": "AWS::Redshift::Cluster",
        "Properties": {
            "ClusterType": "single-node",
            "DBName": "sampledb",
            "MasterUsername": {
                "Fn::Sub": "{{resolve:secretsmanager:${RedshiftSecret}::username}}"
            },
            "MasterUserPassword": {
                "Fn::Sub": "{{resolve:secretsmanager:${RedshiftSecret}::password}}"
            }
        }
    }
}
```
YAML example

RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    DBName: sampledb
    MasterUsername: !Sub '{{resolve:secretsmanager:${RedshiftSecret}::username}}'
    MasterUserPassword: !Sub '{{resolve:secretsmanager:${RedshiftSecret}::password}}'
    NodeType: dc2.large
    PubliclyAccessible: false
    Encrypted: true

CT.REDSHIFT.PR.8 rule specification

# ##################################################################
## Rule Specification ##
# ##################################################################
#
# Rule Identifier:
#   redshift_cluster_encrypted_check
#
# Description:
#   This control checks whether an Amazon Redshift cluster is encrypted.
#
# Reports on:
#   AWS::Redshift::Cluster
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any Amazon Redshift cluster resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Amazon Redshift cluster resource
#     And: 'Encrypted' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an Amazon Redshift cluster resource
#
And: 'Encrypted' has been provided and set to a value other than bool(true)
Then: FAIL
Scenario: 4
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon Redshift cluster resource
And: 'Encrypted' has been provided and set to bool(true)
Then: PASS

Constants
let RESOURCE_TYPE = "AWS::Redshift::Cluster"
let INPUT_DOCUMENT = this

Assignments
let redshift_clusters = Resources.*[ Type == %RESOURCE_TYPE ]

Primary Rules
rule redshift_cluster_encrypted_check when is_cfn_template(%INPUT_DOCUMENT) {
%redshift_clusters not empty {
check(%redshift_clusters.Properties)
<<<
[CT.REDSHIFT.PR.8]: Require an Amazon Redshift cluster to be encrypted
[FIX]: Set the value of the 'Encrypted' property to true.
>>>}
}
rule redshift_cluster_encrypted_check when is_cfn_hook(%INPUT_DOCUMENT, %RESOURCE_TYPE) {
check(%INPUT_DOCUMENT.%RESOURCE_TYPE.resourceProperties)
<<<
[CT.REDSHIFT.PR.8]: Require an Amazon Redshift cluster to be encrypted
[FIX]: Set the value of the 'Encrypted' property to true.
>>>}

Parameterized Rules
rule check(redshift_cluster) {
%redshift_cluster {
# Scenario 2
Encrypted exists

# Scenario 3 and 4
Encrypted == true
}
}

Utility Rules
rule is_cfn_template(doc) {
%doc {
   AWSTemplateFormatVersion exists or
   Resources exists
}
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
   %doc.%RESOURCE_TYPE.resourceProperties exists
}
CT.REDSHIFT.PR.8 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
RedshiftSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: Redshift cluster secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "examplemasterusername"}'
      GenerateStringKey: password
      PasswordLength: 16
      ExcludeCharacters: ''"@\"\n
RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    MasterUsername:
      Fn::Sub: '${resolve:secretsmanager:${RedshiftSecret}::username}'
    MasterUserPassword:
      Fn::Sub: '${resolve:secretsmanager:${RedshiftSecret}::password}'
    NodeType: dc2.large
    PubliclyAccessible: false
    DBName: exampledb
    Encrypted: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
RedshiftSecret:
  Type: AWS::SecretsManager::Secret
  Properties:
    Description: Redshift cluster secret
    GenerateSecretString:
      SecretStringTemplate: '{"username": "examplemasterusername"}'
      GenerateStringKey: password
      PasswordLength: 16
      ExcludeCharacters: ''"@\n
RedshiftCluster:
  Type: AWS::Redshift::Cluster
  Properties:
    ClusterType: single-node
    MasterUsername:
      Fn::Sub: '${resolve:secretsmanager:${RedshiftSecret}::username}'
    MasterUserPassword:
      Fn::Sub: '${resolve:secretsmanager:${RedshiftSecret}::password}'
    NodeType: dc2.large
    PubliclyAccessible: false
    DBName: exampledb
    Encrypted: false
Proactive controls

[CT.REDSHIFT.PR.9] Require that an Amazon Redshift cluster parameter group is configured to use Secure Sockets Layer (SSL) for encryption of data in transit

This control checks whether an Amazon Redshift cluster parameter group is configured to require encryption by means of Secure Sockets Layer (SSL), for data in transit.

- **Control objective:** Encrypt data in transit
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::Redshift::ClusterParameterGroup
- **AWS CloudFormation guard rule:** CT.REDSHIFT.PR.9 rule specification (p. 1236)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.REDSHIFT.PR.9 rule specification (p. 1236)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.REDSHIFT.PR.9 example templates (p. 1238)

**Explanation**

In Amazon Redshift, you can enable encryption of data in transit between an Amazon Redshift cluster and SQL clients over JDBC/ODBC. To support SSL connections, Amazon Redshift creates and installs certificates on each cluster, which are issued by AWS Certificate Manager (ACM).

**Remediation for rule failure**

Set an entry in Parameters with a ParameterName of require_ssl and a ParameterValue of true.

The examples that follow show how to implement this remediation.

**Amazon Redshift cluster parameter group - Example**

An Amazon Redshift cluster parameter group, configured to require encryption of data in transit. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "ClusterParameterGroup": {
      "Type": "AWS::Redshift::ClusterParameterGroup",
      "Properties": {
         "Description": "Example parameter group",
         "ParameterGroupFamily": "redshift-1.0",
         "Parameters": [
            {
               "ParameterName": "require_ssl",
               "ParameterValue": true
            }
         ]
      }
   }
}
```

**YAML example**

```yaml
---

"ClusterParameterGroup": {
  "Type": "AWS::Redshift::ClusterParameterGroup",
  "Properties": {
    "Description": "Example parameter group",
    "ParameterGroupFamily": "redshift-1.0",
    "Parameters": [
      {
        "ParameterName": "require_ssl",
        "ParameterValue": true
      }
    ]
  }
}
```
ClusterParameterGroup:
  Type: AWS::Redshift::ClusterParameterGroup
Properties:
  Description: Example parameter group
  ParameterGroupFamily: redshift-1.0
  Parameters:
    - ParameterName: require_ssl
      ParameterValue: true

CT.REDSHIFT.PR.9 rule specification

# ###################################################################
##       Rule Specification        
# ###################################################################
#
# Rule Identifier:
#  redshift_parameter_group_require_tls_ssl_check
#
# Description:
#  This control checks whether an Amazon Redshift cluster parameter group is configured to
#  require encryption by means of Secure Sockets Layer (SSL), for data in transit.
#
# Reports on:
#  AWS::Redshift::ClusterParameterGroup
#
# Evaluates:
#  AWS CloudFormation, AWS CloudFormation hook
# #
# Rule Parameters:
#  # None
#
# Scenarios:
#  # Scenario: 1
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#    And: The input document does not contain any Amazon Redshift cluster parameter
group resources
#    Then: SKIP
#  # Scenario: 2
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#    And: The input document contains an Amazon Redshift cluster parameter group
resource
#    And: 'Parameters' has not been provided or has been provided as an empty list
#    Then: FAIL
#  # Scenario: 3
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#    And: The input document contains an Amazon Redshift cluster parameter group
resource
#    And: 'Parameters' has been provided as a non-empty list that does not contain an
#      entry with 'ParameterName' set to 'require_ssl' and 'ParameterValue' set to bool(true) or
#      a supported boolean string value ('true', 'True', 'TRUE', 'on', 'On' or 'ON')
#    Then: FAIL
#  # Scenario: 4
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document

1236
And: The input document contains an Amazon Redshift cluster parameter group resource
And: Any entry in 'Parameters' with a 'ParameterName' is set to 'require_ssl' has a corresponding 'ParameterValue' set to a value other than bool(true) or a supported boolean string value ('true', 'True', 'TRUE', 'on', 'On' or 'ON')
Then: FAIL

Scenario: 5
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document contains an Amazon Redshift cluster parameter group resource
And: 'Parameters' has been provided as a non-empty list
And: 'Parameters' contains an entry with 'ParameterName' set to 'require_ssl' and 'ParameterValue' set to bool(true) or a supported boolean string value ('true', 'True', 'TRUE', 'on', 'On' or 'ON')
And: All entries in 'Parameters' with a 'ParameterName' set to 'require_ssl' have a corresponding 'ParameterValue' set to bool(true) or a supported boolean string value ('true', 'True', 'TRUE', 'on', 'On' or 'ON')
Then: PASS

# Constants

let REDSHIFT_CLUSTER_PARAMETER_GROUP_TYPE = "AWS::Redshift::ClusterParameterGroup"
let REDSHIFT_SUPPORTED_TRUE_VALUES = [ true, /^(true|True|TRUE|on|On|ON)$/ ]
let INPUT_DOCUMENT = this

# Assignments

let redshift_cluster_parameter_groups = Resources.[ Type == %REDSHIFT_CLUSTER_PARAMETER_GROUP_TYPE ]

# Primary Rules

rule redshift_parameter_group_require_tls_ssl_check when is_cfn_template(%INPUT_DOCUMENT)

%redshift_cluster_parameter_groups
not empty {
  check(%redshift_cluster_parameter_groups.Properties)
  \[CT.REDSHIFT.PR.9\]: Require that an Amazon Redshift cluster parameter group is configured to use Secure Sockets Layer (SSL) for encryption of data in transit
  [FIX]: Set an entry in 'Parameters' with a 'ParameterName' of 'require_ssl' and a 'ParameterValue' of true.
}

rule redshift_parameter_group_require_tls_ssl_check when is_cfn_hook(%INPUT_DOCUMENT, %REDSHIFT_CLUSTER_PARAMETER_GROUP_TYPE) {
  check(%INPUT_DOCUMENT.%REDSHIFT_CLUSTER_PARAMETER_GROUP_TYPE.resourceProperties)
  \[CT.REDSHIFT.PR.9\]: Require that an Amazon Redshift cluster parameter group is configured to use Secure Sockets Layer (SSL) for encryption of data in transit
  [FIX]: Set an entry in 'Parameters' with a 'ParameterName' of 'require_ssl' and a 'ParameterValue' of true.
}

# # Parameterized Rules

rule check(redshift_cluster_parameter_groups) {
%redshift_cluster_parameter_groups {
    # Scenario 2
    Parameters exists
    Parameters is_list
    Parameters not empty

    # Scenarios 3, 4 and 5
    some Parameters[*] {
        ParameterName exists
        ParameterValue exists
        ParameterName == "require_ssl"
        ParameterValue in %REDSHIFT_SUPPORTED_TRUE_VALUES
    }
    Parameters [
        ParameterName exists
        ParameterName == "require_ssl"
    ] {
        ParameterValue exists
        ParameterValue in %REDSHIFT_SUPPORTED_TRUE_VALUES
    }
}

# Utility Rules
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.REDSHIFT.PR.9 example templates
You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
ClusterParameterGroup:
    Type: AWS::Redshift::ClusterParameterGroup
    Properties:
        Description: Example parameter group
        ParameterGroupFamily: redshift-1.0
        Parameters:
            - ParameterName: require_ssl
              ParameterValue: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
Amazon Simple Storage Service (Amazon S3) controls

Topics

- [CT.S3.PR.1] Require an Amazon S3 bucket to have block public access settings configured (p. 1239)
- [CT.S3.PR.2] Require an Amazon S3 bucket to have server access logging configured (p. 1243)
- [CT.S3.PR.3] Require an Amazon S3 buckets to have versioning configured and a lifecycle policy (p. 1247)
- [CT.S3.PR.4] Require an Amazon S3 bucket to have event notifications configured (p. 1251)
- [CT.S3.PR.5] Require that an Amazon S3 bucket does not manage user access with an access control list (ACL) (p. 1255)
- [CT.S3.PR.6] Require an Amazon S3 bucket to have lifecycle policies configured (p. 1258)
- [CT.S3.PR.8] Require that Amazon S3 bucket requests use Secure Socket Layer (p. 1262)
- [CT.S3.PR.9] Require that an Amazon S3 bucket has S3 Object Lock activated (p. 1269)
- [CT.S3.PR.10] Require an Amazon S3 bucket to have server-side encryption configured using an AWS KMS key (p. 1272)
- [CT.S3.PR.11] Require an Amazon S3 bucket to have versioning enabled (p. 1276)

[CT.S3.PR.1] Require an Amazon S3 bucket to have block public access settings configured

This control checks whether your Amazon Simple Storage Service (Amazon S3) bucket has a bucket-level Block Public Access (BPA) configuration.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::S3::Bucket
- **AWS CloudFormation guard rule:** [CT.S3.PR.1 rule specification](p. 1240)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.S3.PR.1 rule specification](p. 1240)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.S3.PR.1 example templates](p. 1242)

Explanation

Block Public Access at the Amazon S3 bucket level provides controls to ensure that objects never have public access. Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both.

Unless you intend to have your S3 buckets publicly accessible, you should configure the bucket level Amazon S3 Block Public Access feature.
Usage considerations

- This control is incompatible with Amazon S3 buckets that require a public access configuration.

Remediation for rule failure

The parameters `BlockPublicAcls`, `BlockPublicPolicy`, `IgnorePublicAcls`, `RestrictPublicBuckets` must be set to true under the bucket-level `PublicAccessBlockConfiguration`.

The examples that follow show how to implement this remediation.

Amazon S3 Bucket - Example

Amazon S3 bucket with a bucket level Block Public Access configuration that ensures objects never have public access. The example is shown in JSON and in YAML.

**JSON example**

```json
{
"S3Bucket": {
  "Type": "AWS::S3::Bucket",
  "Properties": {
    "PublicAccessBlockConfiguration": {
      "BlockPublicAcls": true,
      "BlockPublicPolicy": true,
      "IgnorePublicAcls": true,
      "RestrictPublicBuckets": true
    }
  }
}
```

**YAML example**

```
S3Bucket:
  Type: AWS::S3::Bucket
  Properties:
    PublicAccessBlockConfiguration:
      BlockPublicAcls: true
      BlockPublicPolicy: true
      IgnorePublicAcls: true
      RestrictPublicBuckets: true
```

CT.S3.PR.1 rule specification

```
# ###################################
##       Rule Specification        ##
####################################
#
# Rule Identifier:
#   s3_bucket_level_public_access_prohibited_check
#
# Description:
```
# Checks whether Amazon Simple Storage Service (Amazon S3) buckets have a bucket-level Block Public Access (BPA)
# configuration.
#
# Reports on:
# AWS::S3::Bucket
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document does not contain any Amazon S3 bucket resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket resource
# And: 'PublicAccessBlockConfiguration' has not been provided
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket resource
# And: 'PublicAccessBlockConfiguration' has been provided
# And: 'BlockPublicAcls' or 'BlockPublicPolicy' or 'IgnorePublicAcls' or
# 'RestrictPublicBuckets'
# have not been provided
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket Resource
# And: 'PublicAccessBlockConfiguration' has been provided
# And: Any of 'BlockPublicAcls' or 'BlockPublicPolicy' or 'IgnorePublicAcls' or
# 'RestrictPublicBuckets'
# have been set to a value other than bool(true) (e.g. bool(false), str(false),
# other)
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket Resource
# And: 'PublicAccessBlockConfiguration' has been provided
# And: 'BlockPublicAcls' or 'BlockPublicPolicy' or 'IgnorePublicAcls' or
# 'RestrictPublicBuckets'
# have all been set to bool(true)
# Then: PASS
#
# Constants
#
let S3_BUCKET_TYPE = "AWS::S3::Bucket"
let INPUT_DOCUMENT = this
#
# Assignments
#
let s3_buckets = Resources.*[ Type == %S3_BUCKET_TYPE ]
#
# Primary Rules
#
rule s3_bucket_level_public_access_prohibited_check when is_cfn_template(%INPUT_DOCUMENT) %s3_buckets not empty { check(%s3_buckets.Properties)
Proactive controls

[CT.S3.PR.1]: Require an Amazon S3 bucket to have block public access settings configured

[FIX]: The parameters 'BlockPublicAcl', 'BlockPublicPolicy', 'IgnorePublicAcl', 'RestrictPublicBuckets' must be set to true under the bucket-level 'PublicAccessBlockConfiguration'.

rule s3_bucket_level_public_access_prohibited_check when is_cfn_hook(%INPUT_DOCUMENT, %S3_BUCKET_TYPE) {
  check(%INPUT_DOCUMENT.%S3_BUCKET_TYPE.resourceProperties) =>
  [CT.S3.PR.1]: Require an Amazon S3 bucket to have block public access settings configured
  [FIX]: The parameters 'BlockPublicAcl', 'BlockPublicPolicy', 'IgnorePublicAcl', 'RestrictPublicBuckets' must be set to true under the bucket-level 'PublicAccessBlockConfiguration'.
}

# Parameterized Rules
#
rule check(s3_bucket) {
  s3_bucket {
    # Scenario 2
    PublicAccessBlockConfiguration exists
    PublicAccessBlockConfiguration is_struct
    PublicAccessBlockConfiguration {
      # Scenario 3
      BlockPublicAcl exists
      BlockPublicPolicy exists
      IgnorePublicAcl exists
      RestrictPublicBuckets exists
      # Scenarios 4 and 5
      BlockPublicAcl == true
      BlockPublicPolicy == true
      IgnorePublicAcl == true
      RestrictPublicBuckets == true
    }
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  doc {
    AWSTemplateFormatVersion exists or Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.S3.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.
PASS Example - Use this template to verify a compliant resource creation.

<table>
<thead>
<tr>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3Bucket:</td>
</tr>
<tr>
<td>Type: AWS::S3::Bucket</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>PublicAccessBlockConfiguration:</td>
</tr>
<tr>
<td>BlockPublicAcls: true</td>
</tr>
<tr>
<td>BlockPublicPolicy: true</td>
</tr>
<tr>
<td>IgnorePublicAcls: true</td>
</tr>
<tr>
<td>RestrictPublicBuckets: true</td>
</tr>
</tbody>
</table>

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

<table>
<thead>
<tr>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3Bucket:</td>
</tr>
<tr>
<td>Type: AWS::S3::Bucket</td>
</tr>
<tr>
<td>Properties:</td>
</tr>
<tr>
<td>PublicAccessBlockConfiguration:</td>
</tr>
<tr>
<td>BlockPublicAcls: false</td>
</tr>
<tr>
<td>BlockPublicPolicy: false</td>
</tr>
<tr>
<td>IgnorePublicAcls: false</td>
</tr>
<tr>
<td>RestrictPublicBuckets: false</td>
</tr>
</tbody>
</table>

**[CT.S3.PR.2] Require an Amazon S3 bucket to have server access logging configured**

This control checks whether server access logging is enabled for your Amazon S3 bucket.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::S3::Bucket
- **AWS CloudFormation guard rule:** [CT.S3.PR.2 rule specification (p. 1244)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.S3.PR.2 rule specification (p. 1244)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.S3.PR.2 example templates (p. 1246)]

**Explanation**

Server access logging provides detailed records of requests made to a bucket. Server access logs can assist in security and access audits.

**Remediation for rule failure**

Set a LoggingConfiguration on the S3 bucket and optionally set DestinationBucketName to an S3 bucket configured to receive S3 Access Logs.
The examples that follow show how to implement this remediation.

**Amazon S3 Bucket - Example**

Amazon S3 bucket with a server access logging configuration. The example is shown in JSON and in YAML.

**JSON example**

```
{
    "S3Bucket": {
        "Type": "AWS::S3::Bucket",
        "Properties": {
            "LoggingConfiguration": {
                "DestinationBucketName": {
                    "Ref": "LoggingBucket"
                }
            }
        }
    }
}
```

**YAML example**

```
S3Bucket:
  Type: AWS::S3::Bucket
  Properties:
    LoggingConfiguration:
      DestinationBucketName: !Ref 'LoggingBucket'
```

**CT.S3.PR.2 rule specification**

```
# ####################################################################
##       Rule Specification        ##
####################################################################
#
# Rule Identifier:
#   s3_bucket_logging_enabled_check
#
# Description:
#   This control checks whether server access logging is enabled for Amazon S3 buckets.
#
# Reports on:
#   AWS::S3::Bucket
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#     And: The input document does not contain any Amazon S3 bucket resources
#     Then: SKIP
#   Scenario: 2
```
# Scenario: 3
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket resource
# And: 'LoggingConfiguration' has not been provided
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket resource
# And: 'LoggingConfiguration' has been provided
# And: 'LoggingConfiguration.DestinationbucketName' has been provided with an empty string or non-valid local reference
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket resource
# And: 'LoggingConfiguration' has been provided
# Then: PASS
# And: 'LoggingConfiguration.DestinationBucketName' has been provided with a non-empty string or valid local reference
# Then: PASS

# Constants
let S3_BUCKET_TYPE = "AWS::S3::Bucket"
let INPUT_DOCUMENT = this

# Assignments
let s3_buckets = Resources.*[ Type == %S3_BUCKET_TYPE ]

# Primary Rules
# Parameterized Rules
rule s3_bucket_logging_enabled_check when is_cfn_template(%INPUT_DOCUMENT) {
  %s3_buckets not empty {
    check(%s3_buckets.Properties)
    %s3_buckets not empty {
      [CT.S3.PR.2]: Require an Amazon S3 bucket to have server access logging configured
      [FIX]: Set a 'LoggingConfiguration' on the S3 Bucket and optionally set 'DestinationBucketName' to an S3 bucket configured to receive S3 Access Logs.
    >>
  }
}
rule s3_bucket_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %S3_BUCKET_TYPE) {
  check(%INPUT_DOCUMENT.%S3_BUCKET_TYPE.resourceProperties)
  %s3_buckets resourceProperties {
    [CT.S3.PR.2]: Require an Amazon S3 bucket to have server access logging configured
    [FIX]: Set a 'LoggingConfiguration' on the S3 bucket and optionally set 'DestinationBucketName' to an S3 bucket configured to receive S3 Access Logs.
  >>
}

# Parameterized Rules
rule check(s3_bucket) {
  %s3_bucket {
    # Scenario 2 and 4
    LoggingConfiguration exists
LoggingConfiguration is_struct

LoggingConfiguration {
  when DestinationBucketName exists {
    # Scenario 3, 4 and 5
    check_is_string_and_not_empty(DestinationBucketName) or
    check_local_references(%INPUT_DOCUMENT, DestinationBucketName,
    %S3_BUCKET_TYPE)
  }
}

# Utility Rules
#
rule check_is_string_and_not_empty(value) {
  %value {
    this is_string
    this != /\A\s*\z/
  }
}

rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

rule check_local_references(doc, reference_properties, referenced_resource_type) {
  %reference_properties {
    'Fn::GetAtt' {
      query_for_resource(%doc, this[0], %referenced_resource_type)
      <<Local Stack reference was invalid>>
    } or Ref {
      query_for_resource(%doc, this, %referenced_resource_type)
      <<Local Stack reference was invalid>>
    }
  }
}

rule query_for_resource(doc, resource_key, referenced_resource_type) {
  let referenced_resource = %doc.Resources[ keys == %resource_key ]
  %referenced_resource not empty
  %referenced_resource {
    Type == %referenced_resource_type
  }
}

CT.S3.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
[CT.S3.PR.3] Require an Amazon S3 buckets to have versioning configured and a lifecycle policy

This control checks whether your Amazon Simple Storage Service (Amazon S3) version-enabled bucket has a lifecycle policy configured.

- **Control objective:** Optimize costs
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::S3::Bucket
- **AWS CloudFormation guard rule:** [CT.S3.PR.3 rule specification (p. 1248)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.S3.PR.3 rule specification (p. 1248)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.S3.PR.3 example templates (p. 1250)]

**Explanation**

We recommend that you configure lifecycle rules on your Amazon S3 bucket, because these rules help you define actions that you want Amazon S3 to take during an object's lifetime.

- **Usage considerations**
  - This control applies only to Amazon S3 buckets with versioning enabled.

**Remediation for rule failure**

Configure versioning-enabled buckets with at least one active lifecycle rule.

The examples that follow show how to implement this remediation.

**Amazon S3 Bucket - Example One**

Amazon S3 bucket with versioning enabled and an active lifecycle rule. The example is shown in JSON and in YAML.

**JSON example**
```json
{
    "S3Bucket": {
        "Type": "AWS::S3::Bucket",
        "Properties": {
            "VersioningConfiguration": {
                "Status": "Enabled"
            },
            "LifecycleConfiguration": {
                "Rules": [
                    {
                        "Status": "Enabled",
                        "ExpirationInDays": 1,
                        "Id": "FirstRule"
                    }
                ]
            }
        }
    }
}
```

### YAML example

S3Bucket:
  Type: AWS::S3::Bucket
  Properties:
    VersioningConfiguration:
      Status: Enabled
    LifecycleConfiguration:
      Rules:
      - Status: Enabled
        ExpirationInDays: 1
        Id: FirstRule

### CT.S3.PR.3 rule specification

```
# ###################################################################
##   Rule Specification   ##
# ###################################################################
#
# Rule Identifier:
#   s3_version_lifecycle_policy_check
#
# Description:
#   Checks whether Amazon Simple Storage Service (Amazon S3) version-enabled buckets have lifecycle policy configured.
#
# Reports on:
#   AWS::S3::Bucket
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
```
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document does not contain any Amazon S3 bucket resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket Resource
# And: The S3 bucket does not have versioning enabled (VersioningConfiguration is
# missing or
# VersioningConfiguration.Status is set to Suspended)
# Then: SKIP
# Scenario: 3
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket Resource
# And: The S3 bucket has versioning enabled (VersioningConfiguration.Status is set to
# 'Enabled')
# And: 'LifecycleConfiguration' has been been provided and there are no 'Rules' with
# 'Status' set to 'Enabled'
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket Resource
# And: The S3 bucket has versioning enabled (VersioningConfiguration.Status is set to
# 'Enabled')
# And: 'LifecycleConfiguration' has been been provided and there is at least one
# 'Rule' with 'Status' set to
# 'Enabled' in the 'LifecycleConfiguration'
# Then: PASS

## Constants

let S3_BUCKET_TYPE = "AWS::S3::Bucket"
let INPUT_DOCUMENT = this

## Assignments

let s3_buckets = Resources.*[ Type == %S3_BUCKET_TYPE ]

## Primary Rules

rule s3_version_lifecycle_policy_check when is_cfn_template(%INPUT_DOCUMENT)
  %s3_buckets not empty {
    check(%s3_buckets.Properties)
    <<
    [CT.S3.PR.3]: Require an Amazon S3 buckets to have versioning configured and a
    lifecycle policy
    [FIX]: Configure versioning-enabled buckets with at least one active lifecycle
    rule.
    >>
  }

rule s3_version_lifecycle_policy_check when is_cfn_hook(%INPUT_DOCUMENT, %S3_BUCKET_TYPE) {
  check(%INPUT_DOCUMENT.%S3_BUCKET_TYPE.resourceProperties)
  <<
  [CT.S3.PR.3]: Require an Amazon S3 buckets to have versioning configured and a
  lifecycle policy
  [FIX]: Configure versioning-enabled buckets with at least one active lifecycle
  rule.
  >>
}

## Parameterized Rules
# rule check(s3_bucket)
% s3_bucket [  
    filter_s3_buckets_with_versioning_enabled(this)  
] {  
    # Scenario 2  
    LifecycleConfiguration exists  
    LifecycleConfiguration is_struct  
    LifecycleConfiguration {  
        # Scenario 3 and 4  
        Rules exists  
        Rules is_list  
        Rules not empty  
        some Rules[*] {  
            Status exists  
            Status == "Enabled"  
        }  
    }  
}  

rule filter_s3_buckets_with_versioning_enabled(s3_bucket) {  
    % s3_bucket {  
        VersioningConfiguration exists  
        VersioningConfiguration is_struct  
        VersioningConfiguration {  
            Status exists  
            Status == "Enabled"  
        }  
    }  
}  

# Utility Rules  
# rule is_cfn_template(doc) {  
% doc {  
    AWSTemplateFormatVersion exists or  
    Resources exists  
}  
}  

rule is_cfn_hook(doc, RESOURCE_TYPE) {  
% doc.%RESOURCE_TYPE.resourceProperties exists  
}  

CT.S3.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:  
Bucket:  
    Type: AWS::S3::Bucket  
Properties:  
    VersioningConfiguration:  
        Status: Enabled
**LifecycleConfiguration:**
- **Status:** Enabled
  - **ExpirationInDays:** 1
- **Id:** FirstRule

**FAIL Example** - Use this template to verify that the control prevents non-compliant resource creation.

```yaml
Resources:
  Bucket:
    Type: AWS::S3::Bucket
    Properties:
      VersioningConfiguration:
        Status: Enabled
      LifecycleConfiguration:
        Rules:
        - Status: Disabled
          ExpirationInDays: 1
          Id: FirstRule
```

[**CT.S3.PR.4**] **Require an Amazon S3 bucket to have event notifications configured**

This control checks whether Amazon S3 events notifications are enabled on your Amazon S3 bucket.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::S3::Bucket
- **AWS CloudFormation guard rule:** [CT.S3.PR.4 rule specification (p. 1253)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: [CT.S3.PR.4 rule specification (p. 1253)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.S3.PR.4 example templates (p. 1255)]

**Explanation**

By enabling event notifications, you receive alerts on your Amazon S3 buckets when specific events occur. For example, you can be notified of object creation, object removal, and object restoration. These notifications can alert relevant teams to accidental or intentional modifications that may lead to unauthorized data access.

**Remediation for rule failure**

Set a `NotificationConfiguration` parameter on your bucket with one of `EventBridgeConfiguration`, `LambdaConfigurations`, `QueueConfigurations` or `TopicConfigurations`.

The examples that follow show how to implement this remediation.
Amazon S3 Bucket - Example One

Amazon S3 bucket with Amazon EventBridge notifications configured. The example is shown in JSON and in YAML.

**JSON example**

```
{
  "S3Bucket": {
    "Type": "AWS::S3::Bucket",
    "Properties": {
      "NotificationConfiguration": {
        "EventBridgeConfiguration": {
          "EventBridgeEnabled": true
        }
      }
    }
  }
}
```

**YAML example**

```
S3Bucket:
  Type: AWS::S3::Bucket
  Properties:
    NotificationConfiguration:
      EventBridgeConfiguration:
        EventBridgeEnabled: true
```

The examples that follow show how to implement this remediation.

Amazon S3 Bucket - Example Two

Amazon S3 bucket with SNS topic notifications configured. The example is shown in JSON and in YAML.

**JSON example**

```
{
  "S3Bucket": {
    "Type": "AWS::S3::Bucket",
    "Properties": {
      "NotificationConfiguration": {
        "TopicConfigurations": [
          {
            "Topic": {
              "Ref": "SnsTopic"
            },
            "Event": "s3:ReducedRedundancyLostObject"
          }
        ]
      }
    }
  }
}
```

**CT.S3.PR.4 rule specification**

```yaml
# This control checks whether Amazon S3 event notifications are enabled on an S3 bucket.
# Reports on: AWS::S3::Bucket
# Evaluates: AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
# None

# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document does not contain any Amazon S3 bucket resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket resource
# And: 'NotificationConfiguration' has not been provided
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket resource
# And: 'NotificationConfiguration' has been provided
# And: At least one of 'EventBridgeConfiguration.EventBridgeEnabled', 'LambdaConfigurations',
# 'QueueConfigurations', or 'TopicConfigurations' have not been provided or provided as empty lists.
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or CloudFormation hook document
# And: The input document contains an Amazon S3 bucket resource
# And: 'NotificationConfiguration' has been provided
# And: 'EventBridgeConfiguration.EventBridgeEnabled' is set to bool(true) or 'LambdaConfigurations',
# 'QueueConfigurations', or 'TopicConfigurations' have been provided with at least one configuration
# Then: PASS
```
### Constants

```
let S3_BUCKET_TYPE = "AWS::S3::Bucket"
let INPUT_DOCUMENT = this
```

### Assignments

```
let s3_buckets = Resources.*[ Type == %S3_BUCKET_TYPE ]
```

### Primary Rules

```
# Primary Rules
#
rule s3_event_notifications_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
    %s3_buckets not empty {
        check(%s3_buckets.Properties)
        <<
        [CT.S3.PR.4]: Require an Amazon S3 bucket to have event notifications configured
        [FIX]: Set a 'NotificationConfiguration' parameter on your bucket with one
        of 'EventBridgeConfiguration', 'LambdaConfigurations', 'QueueConfigurations' or
        'TopicConfigurations.'
        >>
    }
rule s3_event_notifications_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %S3_BUCKET_TYPE) {
    check(%INPUT_DOCUMENT.%S3_BUCKET_TYPE.resourceProperties)
    <<
    [CT.S3.PR.4]: Require an Amazon S3 bucket to have event notifications configured
    [FIX]: Set a 'NotificationConfiguration' parameter on your bucket with one
    of 'EventBridgeConfiguration', 'LambdaConfigurations', 'QueueConfigurations' or
    'TopicConfigurations.'
    >>
}
```

### Parameterized Rules

```
# Parameterized Rules
#
rule check(s3_bucket) {
    %s3_bucket {
        # Scenario 2
        NotificationConfiguration exists
        NotificationConfiguration is_struct
        NotificationConfiguration {
            # Scenario 3 and 4
            EventBridgeConfiguration exists or
            LambdaConfigurations exists or
            QueueConfigurations exists or
            TopicConfigurations exists
            check_event_bridge_notifications(EventBridgeConfiguration) or
            check_other_notifications(LambdaConfigurations) or
            check_other_notifications(QueueConfigurations) or
            check_other_notifications(TopicConfigurations)
        }
    }
rule check_event_bridge_notifications(configuration) {
    %configuration {
        this is_struct
        EventBridgeEnabled exists
        EventBridgeEnabled == true
    }
```

---

1254
CT.S3.PR.4 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
Bucket:
  Type: AWS::S3::Bucket
  Properties:
    NotificationConfiguration:
      EventBridgeConfiguration:
        EventBridgeEnabled: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
Bucket:
  Type: AWS::S3::Bucket
  Properties:
    NotificationConfiguration: {}

[CT.S3.PR.5] Require that an Amazon S3 bucket does not manage user access with an access control list (ACL)

This control checks whether your Amazon Simple Storage Service (Amazon S3) bucket allows user permissions through access control lists.

- **Control objective:** Enforce least privilege
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
**Resource types**: AWS::S3::Bucket

**AWS CloudFormation guard rule**: [CT.S3.PR.5 rule specification](p. 1256)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: [CT.S3.PR.5 rule specification](p. 1256)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.S3.PR.5 example templates](p. 1258)

**Explanation**

ACLs are legacy access control mechanisms that predate IAM. Instead of ACLs, we recommend using IAM policies or Amazon S3 bucket policies to more easily manage access to your S3 buckets.

**Remediation for rule failure**

Manage access to Amazon S3 buckets with bucket resource policies and IAM identity policies instead.

The examples that follow show how to implement this remediation.

**Amazon S3 Bucket - Example**

Amazon S3 bucket that does not allow user permissions through access control lists by omitting the `AccessControl` property. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "S3Bucket": {
      "Type": "AWS::S3::Bucket",
      "Properties": {}
   }
}
```

**YAML example**

```yaml
S3Bucket:
  Type: AWS::S3::Bucket
  Properties: {}
```

**CT.S3.PR.5 rule specification**

```bash
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# Rule Specification #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# # Rule Identifier: #
# s3_bucket_acl_prohibited_check #
# # Description: #
# Checks whether Amazon Simple Storage Service (Amazon S3) buckets allow user permissions through access control lists.
```

1256
# Reports on:
#   AWS::S3::Bucket
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#     And: The input document does not contain any Amazon S3 bucket resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#     And: The input document contains an Amazon S3 bucket resource
#     And: 'AccessControl' has been provided on the S3 bucket resource
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or CloudFormation hook document
#     And: The input document contains an Amazon S3 bucket resource
#     And: 'AccessControl' has not been provided on the S3 bucket resource
#     Then: PASS
#
# Constants
#
let S3_BUCKET_TYPE = "AWS::S3::Bucket"
let INPUT_DOCUMENT = this
#
# Assignments
#
let s3_buckets = Resources.*[ Type == %S3_BUCKET_TYPE ]
#
# Primary Rules
#
rule s3_bucket_acl_prohibited_check when is_cfn_template(%INPUT_DOCUMENT) {
  %s3_buckets not empty {
    check(%s3_buckets.Properties)
    <<
    [CT.S3.PR.5]: Require that an Amazon S3 bucket does not manage user access with an access control list (ACL)
    [FIX]: Manage access to Amazon S3 buckets with bucket resource policies and IAM identity policies instead.
    >>
  }
}
rule s3_bucket_acl_prohibited_check when is_cfn_hook(%INPUT_DOCUMENT, %S3_BUCKET_TYPE) {
  check(this.%S3_BUCKET_TYPE.resourceProperties)
  <<
  [CT.S3.PR.5]: Require that an Amazon S3 bucket does not manage user access with an access control list (ACL)
  [FIX]: Manage access to Amazon S3 buckets with bucket resource policies and IAM identity policies instead.
  >>
}
#
# Parameterized Rules
#
rule check(s3_bucket) {
  %s3_bucket {

1257
CT.S3.PR.5 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls. PASS Example - Use this template to verify a compliant resource creation.

Resources:
Bucket:
    Type: AWS::S3::Bucket
    Properties: {}

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
Bucket:
    Type: AWS::S3::Bucket
    Properties:
        AccessControl: Private

[CT.S3.PR.6] Require an Amazon S3 bucket to have lifecycle policies configured

This control checks whether a lifecycle rule is configured for Amazon S3 buckets.

- **Control objective**: Optimize costs, Improve availability
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::S3::Bucket
- **AWS CloudFormation guard rule**: [CT.S3.PR.6 rule specification](p. 1259)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.S3.PR.6 rule specification](p. 1259)
Proactive controls

- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.S3.PR.6 example templates (p. 1261)

Explanation

Configuring lifecycle rules on your Amazon S3 bucket defines actions that you want S3 to take during an object's lifetime. For example, you can transition objects to another storage class, archive them, or delete them after a specified period of time.

Remediation for rule failure

Configure at least one active lifecycle rule in LifecycleConfiguration.Rules by setting Status on a rule to Enabled.

The examples that follow show how to implement this remediation.

Amazon S3 Bucket - Example

Amazon S3 bucket configured with an active lifecycle rule. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "S3Bucket": {
        "Type": "AWS::S3::Bucket",
        "Properties": {
            "LifecycleConfiguration": {
                "Rules": [
                    {
                        "Status": "Enabled",
                        "ExpirationInDays": 1,
                        "Id": "FirstRule"
                    }
                ]
            }
        }
    }
}
```

**YAML example**

```yaml
S3Bucket:
  Type: AWS::S3::Bucket
  Properties:
    LifecycleConfiguration:
      Rules:
      - Status: Enabled
        ExpirationInDays: 1
        Id: FirstRule
```

**CT.S3.PR.6 rule specification**

```bash
# ###################################################################
##       Rule Specification       ##
# ###################################################################
```

1259
# Rule Identifier:
# s3_lifecycle_policy_check
#
# Description:
# This control checks whether a lifecycle rule is configured for Amazon S3 buckets.
#
# Reports on:
# AWS::S3::Bucket
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain any S3 bucket resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an S3 bucket resource
# And: 'LifecycleConfiguration.Rules' has not been provided or has been provided where 'Rules' is an empty list
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an S3 bucket resource
# And: The S3 bucket has versioning enabled (VersioningConfiguration.Status is set to 'Enabled')
# And: 'LifecycleConfiguration.Rules' has been provided as a non-empty list
# And: There are no 'Rules' with 'Status' set to 'Enabled' present in the 'LifecycleConfiguration'
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an S3 bucket resource
# And: The S3 bucket has versioning enabled (VersioningConfiguration.Status is set to 'Enabled')
# And: 'LifecycleConfiguration.Rules' has been provided as a non-empty list
# And: There is at least one entry in 'LifecycleConfiguration.Rules' with 'Status' set to 'Enabled'
# Then: PASS
#
# Constants
#
let S3_BUCKET_TYPE = "AWS::S3::Bucket"
let INPUT_DOCUMENT = this
#
# Assignments
#
let s3_buckets = Resources.[ Type == %S3_BUCKET_TYPE ]
#
# Primary Rules
#
rule s3_lifecycle_policy_check when is_cfn_template(%INPUT_DOCUMENT)
  %s3_buckets not empty {
check(%s3_buckets.Properties)
<<
  [CT.S3.PR.6]: Require an Amazon S3 bucket to have lifecycle policies configured
  [FIX]: Configure at least one active lifecycle rule in
  'LifecycleConfiguration.Rules' by setting 'Status' on a rule to 'Enabled'.
>>
}

rule s3_lifecycle_policy_check when is_cfn_hook(%INPUT_DOCUMENT, %S3_BUCKET_TYPE) {
  check(%INPUT_DOCUMENT.%S3_BUCKET_TYPE.resourceProperties)
<<
  [CT.S3.PR.6]: Require an Amazon S3 bucket to have lifecycle policies configured
  [FIX]: Configure at least one active lifecycle rule in
  'LifecycleConfiguration.Rules' by setting 'Status' on a rule to 'Enabled'.
>>
}

# Parameterized Rules
#
rule check(s3_bucket) {
  %s3_bucket {
    # Scenario 2
    LifecycleConfiguration exists
    LifecycleConfiguration is_struct

    LifecycleConfiguration {
      # Scenario 3 and 4
      Rules exists
      Rules is_list
      Rules not empty

      some Rules[*] {
        Status exists
        Status == "Enabled"
      }
    }
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists  or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.S3.PR.6 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
S3Bucket:
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
S3Bucket:
  Type: AWS::S3::Bucket
  Properties:
    LifecycleConfiguration:
      Rules:
        - Status: Enabled
          ExpirationInDays: 1
          Id: FirstRule

[CT.S3.PR.8] Require that Amazon S3 bucket requests use Secure Socket Layer

This control checks whether Amazon S3 bucket policies require requests to use Secure Socket Layer (SSL).

- Control objective: Encrypt data in transit
- Implementation: AWS CloudFormation Guard Rule
- Control behavior: Proactive
- Resource types: AWS::S3::BucketPolicy
- AWS CloudFormation guard rule: CT.S3.PR.8 rule specification (p. 1264)

Details and examples
- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.S3.PR.8 rule specification (p. 1264)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: CT.S3.PR.8 example templates (p. 1267)

Explanation
Amazon S3 buckets should have policies that require all requests (Action: S3:*) to accept transmission of data over HTTPS in the S3 resource policy only, as indicated by the condition key aws:SecureTransport.

Remediation for rule failure
Configure an Amazon S3 bucket policy statement that denies access to all principals and actions for the S3 bucket and bucket objects when a secure transport protocol is not in use.

The examples that follow show how to implement this remediation.

Amazon S3 Bucket Policy - Example
Amazon S3 bucket policy configured to deny all access to the bucket and bucket objects when transmission of data is not over HTTPS. The example is shown in JSON and in YAML.
**JSON example**

```json
{
  "S3BucketPolicy": {
    "Type": "AWS::S3::BucketPolicy",
    "Properties": {
      "Bucket": {
        "Ref": "S3Bucket"
      },
      "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Deny",
            "Action": "s3:*",
            "Resource": [
              { "Fn::GetAtt": [ "S3Bucket", "Arn" ] },
              "Principal": "*",
              "Condition": {
                "Bool": {
                  "aws:SecureTransport": "false"
                }
              }
            ]
          }
        ]
      }
    }
  }
}
```

**YAML example**

```yaml
S3BucketPolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    Bucket: !Ref 'S3Bucket'
  PolicyDocument:
    Version: 2012-10-17
    Statement:
      - Effect: Deny
        Action: s3:*`
```
Resource:
- !GetAtt 'S3Bucket.Arn'
- !Join
  - ''
  - - !GetAtt 'S3Bucket.Arn'
- /*
Principal: ''
Condition:
  Bool:
    aws:SecureTransport: 'false'

CT.S3.PR.8 rule specification

# ###################################################################
##       Rule Specification        ##
###################################################################
#
# Rule Identifier:
#   s3_bucket_policy_ssl_requests_only_check
#
# Description:
#   This control checks whether Amazon S3 bucket policies require requests to use Secure
#   Socket Layer (SSL).
#
# Reports on:
#   AWS::S3::BucketPolicy
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#     And: The input document does not contain any S3 bucket policies
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#     And: The input document contains an S3 bucket policy
#     And: 'Policydocument' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#     And: The input document contains an S3 bucket policy
#     And: 'Policydocument' does not include a statement that denies Principal ('*',
#          AWS: '*')
#          all Actions ('s3:*', '*') over resource ('*' or bucketArn, bucketObjectArn)
#          when the condition
#          "aws:SecureTransport" is "false"
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
#            document
#     And: The input document contains an S3 bucket policy
#     And: 'Policydocument' includes a statement that denies Principal ('*', AWS: '*')
# all Actions ('s3:*', '*') over resource ('*' or bucketArn, bucketObjectArn) when the condition
# "aws:SecureTransport" is "false"
# Then: PASS

## Constants

let S3_BUCKET_POLICY_TYPE = "AWS::S3::BucketPolicy"
let INPUT_DOCUMENT = this

let S3_BUCKET_ARN_PATTERN = /^arn:aws[a-z0-9-]*:s3:::[a-z0-9][a-z0-9.-]*[a-z0-9]$/
let S3_BUCKET_OBJECT_ARN_PATTERN = /^arn:aws[a-z0-9-]*:s3:::[a-z0-9][a-z0-9.-]*[a-z0-9-]*\/$/

## Assignments

let s3_bucket_policies = Resources.*[ Type == %S3_BUCKET_POLICY_TYPE ]

## Primary Rules

rule s3_bucket_policy_ssl_requests_only_check when is_cfn_template(%INPUT_DOCUMENT) %s3_bucket_policies not empty {
  check(%s3_bucket_policies.Properties)
  <<
  [CT.S3.PR.8]: Require that Amazon S3 buckets request to use Secure Socket Layer
  [FIX]: Configure an Amazon S3 bucket policy statement that denies access to all principals and actions for the S3 bucket and bucket objects when a secure transport protocol is not in use.
  >>
}

rule s3_bucket_policy_ssl_requests_only_check when is_cfn_hook(%INPUT_DOCUMENT, %S3_BUCKET_POLICY_TYPE) {
  check(%INPUT_DOCUMENT.%S3_BUCKET_POLICY_TYPE.resourceProperties)
  <<
  [CT.S3.PR.8]: Require that Amazon S3 buckets request to use Secure Socket Layer
  [FIX]: Configure an Amazon S3 bucket policy statement that denies access to all principals and actions for the S3 bucket and bucket objects when a secure transport protocol is not in use.
  >>
}

## Parameterized Rules

rule check(s3_bucket_policy) {
  %s3_bucket_policy {
  # Scenario 2
  PolicyDocument exists
  PolicyDocument is_struct
  PolicyDocument {
    Statement exists
    Statement is_list or
    Statement is_struct

    # Scenario 3 and 4
    some Statement[*] {
      check_statement_ssl_requests_only(this)
    }
  }
  }
}
rule check_statement_ssl_requests_only(statement) {
  %statement{
    check_all_required_statement_properties(this)
    Effect == "Deny"
    Action[*] in ["s3:*", "+"]
    Principal == "*" or
    Principal {
      AWS exists
      AWS == "*"
    }
    Resource[*] == "*" or
    check_resource_for_bucket_arns(Resource) or
    check_resource_for_bucket_arn_refs(Resource)
    Condition is_struct
    Condition == {
      "Bool": {
        "aws:SecureTransport": "false"
      }
    }
  }
}

rule check_all_required_statement_properties(statement) {
  %statement {
    Effect exists
    Action exists
    Principal exists
    Condition exists
    Resource exists
  }
}

rule check_resource_for_bucket_arns(resource) {
  %resource {
    this is_list
    this not empty
    some this[*] == %S3_BUCKET_ARN_PATTERN
    some this[*] == %S3_BUCKET_OBJECT_ARN_PATTERN
  }
}

rule check_resource_for_bucket_arn_refs(resource) {
  %resource {
    this is_list
    this not empty
    some this[*] {
      check_local_bucket_arn_reference(%INPUT_DOCUMENT, this, "AWS::S3::Bucket")
    }
    some this[*] {
      check_local_bucket_object_arn_reference(%INPUT_DOCUMENT, this, "AWS::S3::Bucket")
    }
  }
}

rule check_local_bucket_arn_reference(doc, reference_properties, referenced_resource_type) {
  %reference_properties {
    'Fn::GetAtt' {
      check_get_att_bucket_arn(this)
rule check_local_bucket_object_arn_reference(doc, reference_properties, referenced_resource_type) {
    %reference_properties {
        'Fn::Join' {
            this is_list
            this not empty
            this[1][0] {
                'Fn::GetAtt' {
                    check_get_att_bucket_arn(this)
                }
            }
            this[1][1] == "/*"
        }
    }
}

rule check_get_att_bucket_arn(get_att){
    %get_att {
        this is_list
        this not empty
        this[1] == "Arn"
        query_for_resource(%doc, this[0], %referenced_resource_type)
        <<Local Stack reference was invalid>>
    }
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

rule query_for_resource(doc, resource_key, referenced_resource_type) {
    let referenced_resource = %doc.Resources[ keys == %resource_key ]
    %referenced_resource not empty
    %referenced_resource {
        Type == %referenced_resource_type
    }
}

CT.S3.PR.8 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
S3Bucket:
    Type: AWS::S3::Bucket
Properties:

PublicAccessBlockConfiguration:
  BlockPublicAcls: true
  BlockPublicPolicy: true
  IgnorePublicAcls: true
  RestrictPublicBuckets: true

S3BucketPolicy:
  Type: AWS::S3::BucketPolicy
  Properties:
    Bucket:
      Ref: S3Bucket
    PolicyDocument:
      Version: 2012-10-17
      Statement:
        - Effect: Deny
          Action: s3:*
          Resource:
            - Fn::GetAtt:
              - S3Bucket
              - Arn
            - Fn::Join:
              - ''
              - - Fn::GetAtt:
                - S3Bucket
                - Arn
              - /*
          Principal: '*'
          Condition:
            Bool:
              aws:SecureTransport: 'false'

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:

S3Bucket:
  Type: AWS::S3::Bucket
  Properties:
    PublicAccessBlockConfiguration:
      BlockPublicAcls: true
      BlockPublicPolicy: true
      IgnorePublicAcls: true
      RestrictPublicBuckets: true
    S3BucketPolicy:
      Type: AWS::S3::BucketPolicy
      Properties:
        Bucket:
          Ref: S3Bucket
        PolicyDocument:
          Version: 2012-10-17
          Statement:
            - Effect: Allow
              Action: s3:*
              Resource:
                - Fn::GetAtt:
                  - S3Bucket
                  - Arn
                - Fn::Join:
                  - ''
                  - - Fn::GetAtt:
                    - S3Bucket
                    - Arn
                  - /*
[CT.S3.PR.9] Require that an Amazon S3 bucket has S3 Object Lock activated

This control checks whether an Amazon Simple Storage Service (Amazon S3) bucket has been configured to use S3 Object Lock.

- **Control objective:** Protect data integrity
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::S3::Bucket
- **AWS CloudFormation guard rule:** CT.S3.PR.9 rule specification (p. 1270)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.S3.PR.9 rule specification (p. 1270)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.S3.PR.9 example templates (p. 1271)

Explanation

S3 Object Lock allows you to store objects using a write-once-read-many (WORM) model. Object Lock can help prevent objects from being deleted or overwritten for a fixed amount of time, or indefinitely. You can use S3 Object Lock to meet regulatory requirements that require WORM storage, or to add an extra layer of protection against object changes and deletion.

Usage considerations

- When you create an Amazon S3 bucket with object lock activated, S3 automatically enables versioning for the bucket.

Remediation for rule failure

Set ObjectLockEnabled to true.

The examples that follow show how to implement this remediation.

S3 bucket - Example

An Amazon S3 bucket configured with S3 object lock enabled. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "Bucket": {
        "Type": "AWS::S3::Bucket",
        ...
    }
}
```
"Properties": {
  "ObjectLockEnabled": true
}
}

YAML example

Bucket:
  Type: AWS::S3::Bucket
  Properties:
    ObjectLockEnabled: true

CT.S3.PR.9 rule specification

# ####################################################################################################
##       Rule Specification                        ##
# ####################################################################################################
#
# Rule Identifier:
#   s3_bucket_object_lock_enabled_check
#
# Description:
#   This control checks whether an Amazon Simple Storage Service (Amazon S3) bucket has been configured to use S3 Object Lock.
#
# Reports on:
#   AWS::S3::Bucket
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any S3 bucket resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an S3 bucket resource
#     And: 'ObjectLockEnabled' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an S3 bucket resource
#     And: 'ObjectLockEnabled' has been provided and set to a value other than bool(true)
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document contains an S3 bucket resource
#     And: 'ObjectLockEnabled' has been provided and set to bool(true)
# Constants

let S3_BUCKET_TYPE = "AWS::S3::Bucket"
let INPUT_DOCUMENT = this

# Assignments

let s3_buckets = Resources.*[ Type == %S3_BUCKET_TYPE ]

# Primary Rules

# rule s3_bucket_object_lock_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
rule s3_bucket_object_lock_enabled_check when is_cfn_template(%INPUT_DOCUMENT)
    %s3_buckets not empty {
        check(%s3_buckets.Properties)
        <<
        [CT.S3.PR.9]: Require that an Amazon S3 bucket has S3 Object Lock activated
        [FIX]: Set 'ObjectLockEnabled' to 'true'.
        >>
    }

rule s3_bucket_object_lock_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %S3_BUCKET_TYPE)
    {
        check(this.%S3_BUCKET_TYPE.resourceProperties)
        <<
        [CT.S3.PR.9]: Require that an Amazon S3 bucket has S3 Object Lock activated
        [FIX]: Set 'ObjectLockEnabled' to 'true'.
        >>
    }

# Parameterized Rules

# rule check(s3_bucket) {
rule check(s3_bucket) {
    %s3_bucket {
        # Scenario 2
        ObjectLockEnabled exists
        # Scenarios 3 and 4
        ObjectLockEnabled == true
    }
}

# Utility Rules

# rule is_cfn_template(doc) {
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.S3.PR.9 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.
PASS Example - Use this template to verify a compliant resource creation.

Resources:
Bucket:
  Type: AWS::S3::Bucket
  Properties:
    ObjectLockEnabled: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
Bucket:
  Type: AWS::S3::Bucket
  Properties:
    ObjectLockEnabled: false

[CT.S3.PR.10] Require an Amazon S3 bucket to have server-side encryption configured using an AWS KMS key

This control checks whether default server-side encryption is enabled on an Amazon S3 bucket using AWS KMS.

- **Control objective:** Encrypt data at rest
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::S3::Bucket
- **AWS CloudFormation guard rule:** [CT.S3.PR.10 rule specification](p. 1273)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.S3.PR.10 rule specification](p. 1273)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.S3.PR.10 example templates](p. 1276)

Explanation

Server-side encryption (SSE) is the encryption of data at its destination by the application or service that receives the data. Unless you specify otherwise, Amazon S3 buckets use SSE-S3 by default to encrypt objects. However, for added control, you can choose to configure buckets to use server-side encryption with AWS KMS keys (SSE-KMS) instead. Amazon S3 encrypts your data at the object level as it writes data to disks in AWS data centers, and then decrypts the data for you, when you require access to it.

Remediation for rule failure


The examples that follow show how to implement this remediation.
S3 Bucket - Example

An Amazon S3 bucket configured with AWS Key Management Service (AWS KMS) (SSE-KMS) default server-side encryption. The example is shown in JSON and in YAML.

JSON example

```json
{
    "S3Bucket": {
        "Type": "AWS::S3::Bucket",
        "Properties": {
            "BucketEncryption": {
                "ServerSideEncryptionConfiguration": [
                    {
                        "ServerSideEncryptionByDefault": {
                            "SSEAlgorithm": "aws:kms"
                        }
                    }
                ]
            }
        }
    }
}
```

YAML example

```yaml
S3Bucket:
    Type: AWS::S3::Bucket
    Properties:
        BucketEncryption:
            ServerSideEncryptionConfiguration:
                - ServerSideEncryptionByDefault:
                    SSEAlgorithm: aws:kms
```

CT.S3.PR.10 rule specification

```yaml
# ###################################################################
# Rule Specification                                              #
# ###################################################################
# # Rule Identifier:  # s3_bucket_default_encryption_kms_check  #
# # Description:     # This control checks whether default server-side encryption is enabled on an Amazon S3 bucket using AWS KMS. #
# # Reports on:      # AWS::S3::Bucket #
# # Evaluates:       # AWS CloudFormation, AWS CloudFormation hook #
# # Rule Parameters: # None #
```
# Scenarios:

## Scenario: 1

- **Given:** The input document is an AWS CloudFormation or AWS CloudFormation hook document
- **And:** The input document does not contain any S3 bucket resources
- **Then:** SKIP

## Scenario: 2

- **Given:** The input document is an AWS CloudFormation or AWS CloudFormation hook document
- **And:** The input document contains an S3 bucket resource
- **And:** 'ServerSideEncryptionConfiguration' in 'BucketEncryption' has not been provided or provided as an empty list
- **Then:** FAIL

## Scenario: 3

- **Given:** The input document is an AWS CloudFormation or AWS CloudFormation hook document
- **And:** The input document contains an S3 bucket resource
- **And:** 'ServerSideEncryptionConfiguration' in 'BucketEncryption' has been provided as a non empty list
- **And:** 'ServerSideEncryptionConfiguration' in 'BucketEncryption' does not contain an encryption rule with a 'ServerSideEncryptionByDefault' configuration
- **Then:** FAIL

## Scenario: 4

- **Given:** The input document is an AWS CloudFormation or AWS CloudFormation hook document
- **And:** The input document contains an S3 bucket resource
- **And:** 'ServerSideEncryptionConfiguration' in 'BucketEncryption' has been provided as a non empty list
- **And:** 'ServerSideEncryptionConfiguration' in 'BucketEncryption' contains an encryption rule with a 'ServerSideEncryptionByDefault' configuration
- **And:** For an encryption rule, 'SSEAlgorithm' in 'ServerSideEncryptionByDefault' is not provided or has been provided and set to an SSE Algorithm other than 'aws:kms' or 'aws:kms:dsse'
- **Then:** FAIL

## Scenario: 5

- **Given:** The input document is an AWS CloudFormation or AWS CloudFormation hook document
- **And:** The input document contains an S3 bucket resource
- **And:** 'ServerSideEncryptionConfiguration' in 'BucketEncryption' has been provided as a non empty list
- **And:** 'ServerSideEncryptionConfiguration' in 'BucketEncryption' contains an encryption rule with a 'ServerSideEncryptionByDefault' configuration
- **And:** For all encryption rules, 'SSEAlgorithm' in 'ServerSideEncryptionByDefault' is provided and set to 'aws:kms' or 'aws:kms:dsse'
- **Then:** PASS

## Constants

- `let S3_BUCKET_TYPE = "AWS::S3::Bucket"
- let AUTHORIZED_SSE_ALGORITHMS = [ "aws:kms", "aws:kms:dsse" ]
- let INPUT_DOCUMENT = this

## Assignments

- `let s3_buckets = Resources.*[ Type == %S3_BUCKET_TYPE ]`

## Primary Rules
# Proactive controls

## Parameterized Rules

```python
# Parameterized Rules

# s3_bucket_default_encryption_kms_check

rule s3_bucket_default_encryption_kms_check when is_cfn_template(%INPUT_DOCUMENT) {
    %s3_buckets not empty {
        check(%s3_buckets.Properties)
        <<
        [CT.S3.PR.10]: Require an Amazon S3 bucket to have server-side encryption configured using an AWS KMS key
        [FIX]: Set an encryption rule in 'BucketEncryption.ServerSideEncryptionConfiguration' with a 'ServerSideEncryptionByDefault.SSEAlgorithm' configuration of 'aws:kms' or 'aws:kms:dsse'
    }
}

rule s3_bucket_default_encryption_kms_check when is_cfn_hook(%INPUT_DOCUMENT, %S3_BUCKET_TYPE) {
    check(%INPUT_DOCUMENT.%S3_BUCKET_TYPE.resourceProperties)
    <<
    [CT.S3.PR.10]: Require an Amazon S3 bucket to have server-side encryption configured using an AWS KMS key
    [FIX]: Set an encryption rule in 'BucketEncryption.ServerSideEncryptionConfiguration' with a 'ServerSideEncryptionByDefault.SSEAlgorithm' configuration of 'aws:kms' or 'aws:kms:dsse'
}
```

## Utility Rules

```python
# Utility Rules

# is_cfn_template

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}
```
CT.S3.PR.10 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  Bucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketEncryption:
        ServerSideEncryptionConfiguration:
          - ServerSideEncryptionByDefault:
            SSEAlgorithm: aws:kms
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
  Bucket:
    Type: AWS::S3::Bucket
    Properties: {}
```

[CT.S3.PR.11] Require an Amazon S3 bucket to have versioning enabled

This control checks whether an Amazon Simple Storage Service (Amazon S3) bucket has versioning enabled.

- **Control objective**: Improve availability
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::S3::Bucket
- **AWS CloudFormation guard rule**: [CT.S3.PR.11 rule specification](p. 1277)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: [CT.S3.PR.11 rule specification](p. 1277)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.S3.PR.11 example templates](p. 1279)

Explanation

Versioning keeps multiple variants of an object in the same Amazon S3 bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your S3 bucket. With versioning, you can recover more easily from unintended user actions and application failures.
Usage considerations

- If you have an unversioned bucket with an object expiration lifecycle configuration, and if you want to maintain the same permanent delete behavior when you enable versioning, you must set an expiration configuration for noncurrent objects. The noncurrent expiration configuration lifecycle manages deletion of noncurrent object versions in the version-enabled bucket. (A version-enabled bucket maintains one current, and zero or more noncurrent, object versions.)

Remediation for rule failure

Set the Status in VersioningConfiguration to Enabled.

The examples that follow show how to implement this remediation.

S3 Bucket - Example One

An Amazon S3 bucket with versioning enabled. The example is shown in JSON and in YAML.

JSON example

```
{
   "Bucket": {
      "Type": "AWS::S3::Bucket",
      "Properties": {
         "VersioningConfiguration": {
            "Status": "Enabled"
         }
      }
   }
}
```

YAML example

```
Bucket:
   Type: AWS::S3::Bucket
   Properties:
      VersioningConfiguration:
         Status: Enabled
```

CT.S3.PR.11 rule specification

```
# ###################################################################
##       Rule Specification       ##
# ###################################################################
#
# Rule Identifier:
#   s3_bucket_versioning_enabled_check
#
# Description:
#   This control checks whether an Amazon Simple Storage Service (Amazon S3) bucket has
#   versioning enabled.
#```
# Reports on:
#   AWS::S3::Bucket
#
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#        And: The input document does not contain any S3 bucket resources
#        Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#        And: The input document contains an S3 bucket resource
#        And: 'VersioningConfiguration' has not been provided
#        Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#        And: The input document contains an S3 bucket resource
#        And: 'VersioningConfiguration' has been provided
#        And: 'Status' in 'VersioningConfiguration' has not been provided or has been
#        and set to a value other than 'Enabled'
#        Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#        And: The input document contains an S3 bucket resource
#        And: 'VersioningConfiguration' has been provided
#        And: 'Status' in 'VersioningConfiguration' has been provided and set to 'Enabled'
#        Then: PASS
#
# Constants
#
let S3_BUCKET_TYPE = "AWS::S3::Bucket"
let INPUT_DOCUMENT = this
#
# Assignments
#
let s3_buckets = Resources.*[ Type == %S3_BUCKET_TYPE ]
#
# Primary Rules
#
rule s3_bucket_versioning_enabled_check when is_cfn_template(%INPUT_DOCUMENT) %s3_buckets not empty {
  check(%s3_buckets.Properties)
  <<
    [CT.S3.PR.11]: Require an Amazon S3 bucket to have versioning enabled
    [FIX]: Set the 'Status' in 'VersioningConfiguration' to Enabled.
  >>
}
rule s3_bucket_versioning_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %S3_BUCKET_TYPE) {
  check(%INPUT_DOCUMENT.%S3_BUCKET_TYPE.resourceProperties)
  <<
    [CT.S3.PR.11]: Require an Amazon S3 bucket to have versioning enabled
  >>
[FIX]: Set the 'Status' in 'VersioningConfiguration' to Enabled.

```
#
# Parameterized Rules
#
rule check(s3_bucket) {
  %s3_bucket {
    # Scenario 2
    VersioningConfiguration exists
    # Scenarios 3 and 4
    VersioningConfiguration is_struct
    VersioningConfiguration {
      Status exists
      Status == "Enabled"
    }
  }
}
```

```
#
# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

CT.S3.PR.11 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
Bucket:
  Type: AWS::S3::Bucket
  Properties:
    VersioningConfiguration:
      Status: Enabled
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
Bucket:
  Type: AWS::S3::Bucket
  Properties: {}
```
Amazon SageMaker controls

Topics
- [CT.SAGEMAKER.PR.1] Require an Amazon SageMaker notebook instance to prevent direct internet access (p. 1280)
- [CT.SAGEMAKER.PR.2] Require Amazon SageMaker notebook instances to be deployed within a custom Amazon VPC (p. 1286)
- [CT.SAGEMAKER.PR.3] Require Amazon SageMaker notebook instances to have root access disallowed (p. 1290)

[CT.SAGEMAKER.PR.1] Require an Amazon SageMaker notebook instance to prevent direct internet access

This control checks that direct internet access is not allowed for an Amazon SageMaker notebook instance.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::SageMaker::NotebookInstance
- **AWS CloudFormation guard rule:** [CT.SAGEMAKER.PR.1 rule specification (p. 1281)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.SAGEMAKER.PR.1 rule specification (p. 1281)]
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.SAGEMAKER.PR.1 example templates (p. 1284)]

Explanation

When you configure your SageMaker notebook instance without a VPC, direct internet access is allowed for your instance, by default. Instead, you should configure your instance with a VPC and change the default setting to **Disable - Access the internet through a VPC.**

To train or host models from a notebook, you require internet access. To set up internet access, make sure that your VPC has a NAT gateway, and that your security group allows outbound connections.

Ensure that access to your SageMaker configuration is limited to authorized users. Restrict users' IAM permissions for modifying SageMaker settings and resources.

**Usage considerations**

- To set up outbound internet access for Amazon SageMaker notebook instances when this control is activated - First, associate the notebook instance with a private subnet that has access to the internet, through a default route to a NAT gateway instance. Also, be sure that the security groups assigned to the notebook instance, and the network access control list (NACL) of the private subnet, allow outbound traffic to the internet.

**Remediation for rule failure**

Set **DirectInternetAccess** to **Disabled** and provide a **SubnetId** and one or more **SecurityGroupIds**.
The examples that follow show how to implement this remediation.

**Amazon SageMaker Notebook Instance - Example**

Amazon SageMaker notebook instance configured with direct internet access deactivated. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "SageMakerNoteBookInstance": {
    "Type": "AWS::SageMaker::NotebookInstance",
    "Properties": {
      "InstanceType": "ml.t2.medium",
      "RoleArn": {
        "Fn::GetAtt": [
          "ExecutionRole",
          "Arn"
        ]
      },
      "DirectInternetAccess": "Disabled",
      "SubnetId": {
        "Ref": "Subnet"
      },
      "SecurityGroupIds": [
        {
          "Fn::GetAtt": [
            "SecurityGroup",
            "GroupId"
          ]
        ]
      }
    }
  }
}
```

**YAML example**

```yaml
SageMakerNoteBookInstance:
  Type: AWS::SageMaker::NotebookInstance
  Properties:
    InstanceType: ml.t2.medium
    RoleArn: !GetAtt 'ExecutionRole.Arn'
    DirectInternetAccess: Disabled
    SubnetId: !Ref 'Subnet'
    SecurityGroupIds:
      - !GetAtt 'SecurityGroup.GroupId'
```

**CT.SAGEMAKER.PR.1 rule specification**

```plaintext
# ###################################################################
##       Rule Specification       ##
# ###################################################################
# Rule Identifier:
# sagemaker_notebook_no_direct_internet_access_check
# 1281
```
# Description:
This control checks that direct internet access is not allowed for an Amazon SageMaker notebook instance.

# Reports on:
AWS::SageMaker::NotebookInstance

# Evaluates:
AWS CloudFormation, AWS CloudFormation hook

# Rule Parameters:
None

# Scenarios:
# Scenario: 1
Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
And: The input document does not contain any SageMaker notebook instance resources
Then: SKIP

# Scenario: 2
Given: The input document contains an SageMaker notebook instance resource
And: 'DirectInternetAccess' has not been provided on the SageMaker notebook instance resource
Then: FAIL

# Scenario: 3
Given: The input document contains an SageMaker notebook instance resource
And: 'DirectInternetAccess' has been provided on the SageMaker notebook instance resource
And: 'DirectInternetAccess' is set to 'Enabled'
Then: FAIL

# Scenario: 4
Given: The input document contains an SageMaker notebook instance resource
And: 'DirectInternetAccess' has been provided on the SageMaker notebook instance resource
And: 'DirectInternetAccess' is set to 'Disabled'
And: 'SecurityGroupIds' have been provided as a non-empty list with non-empty strings or valid local references
And: 'SubnetId' has been provided as an empty string or non-valid local reference
Then: FAIL

# Scenario: 5
Given: The input document contains an SageMaker notebook instance resource
And: 'DirectInternetAccess' has been provided on the SageMaker notebook instance resource
And: 'DirectInternetAccess' is set to 'Disabled'
And: 'SubnetId' has been provided as a non-empty string or valid local reference
And: 'SecurityGroupIds' have been provided as an empty list or a list that contains empty string values or non-valid local references
Then: FAIL

# Scenario: 6
Given: The input document contains an SageMaker notebook instance resource
And: 'DirectInternetAccess' has been provided on the SageMaker notebook instance resource
And: 'DirectInternetAccess' is set to 'Disabled'
And: 'SecurityGroupIds' have been provided as a non-empty list with non-empty strings or valid local references
And: 'SubnetId' has been provided as a non-empty string or valid local reference
Then: PASS

# Constants
#
let SAGEMAKER_NOTEBOOK_INSTANCE_TYPE = "AWS::SageMaker::NotebookInstance"
let INPUT_DOCUMENT = this

# Assignments
#
let sagemaker_notebook_instances = Resources.*[ Type == %SAGEMAKER_NOTEBOOK_INSTANCE_TYPE ]

# Primary Rules
#
rule sagemaker_notebook_no_direct_internet_access_check when is_cfn_template(this)
    %sagemaker_notebook_instances
not empty {
    check(%sagemaker_notebook_instances.Properties)
    <<
    [CT.SAGEMAKER.PR.1]: Require an Amazon SageMaker notebook instance to prevent
direct internet access
    [FIX]: Set 'DirectInternetAccess' to 'Disabled' and provide a 'SubnetId' and one or
more 'SecurityGroupIds'.
    >>
}

rule sagemaker_notebook_no_direct_internet_access_check when is_cfn_hook(%INPUT_DOCUMENT, %SAGEMAKER_NOTEBOOK_INSTANCE_TYPE) {
    check(%INPUT_DOCUMENT.%SAGEMAKER_NOTEBOOK_INSTANCE_TYPE.resourceProperties)
    <<
    [CT.SAGEMAKER.PR.1]: Require an Amazon SageMaker notebook instance to prevent
direct internet access
    [FIX]: Set 'DirectInternetAccess' to 'Disabled' and provide a 'SubnetId' and one or
more 'SecurityGroupIds'.
    >>
}

# Parameterized Rules
#
rule check(sagemaker_notebook_instances) {
    %sagemaker_notebook_instances {
        # Scenario 2
        DirectInternetAccess exists

        # Scenario 3
        DirectInternetAccess is_string
        DirectInternetAccess == "Disabled"

        # Scenario 4,5 and 6
        check_is_string_and_not_empty(SubnetId) or
        check_local_references(%INPUT_DOCUMENT, SubnetId, "AWS::EC2::Subnet")

        SecurityGroupIds exists
        SecurityGroupIds is_list
        SecurityGroupIds not empty

        SecurityGroupIds[*] {
            check_is_string_and_not_empty(this) or
            check_local_references(%INPUT_DOCUMENT, this, "AWS::EC2::SecurityGroup")
        }
    }
}
# Utility Rules

## rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or 
        Resources exists
    }
}

## rule is_cfn_hook(doc, SAGEMAKER_NOTEBOOK_INSTANCE_TYPE) {
    %doc.%SAGEMAKER_NOTEBOOK_INSTANCE_TYPE.resourceProperties exists
}

## rule check_is_string_and_not_empty(value) {
    %value {
        this is_string
        this != /\A\s*\z/
    }
}

## rule check_local_references(doc, reference_properties, referenced_resource_type) {
    %reference_properties {
        'Fn::GetAtt' {
            query_for_resource(%doc, this[0], %referenced_resource_type)
            <<Local Stack reference was invalid>>
        } or Ref {
            query_for_resource(%doc, this, %referenced_resource_type)
            <<Local Stack reference was invalid>>
        }
    }
}

## rule query_for_resource(doc, resource_key, referenced_resource_type) {
    let referenced_resource = %doc.Resources[ keys == %resource_key ]
    %referenced_resource not empty
    %referenced_resource {
        Type == %referenced_resource_type
    }
}

### CT.SAGEMAKER.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

**Resources:**

**VPC:**
  Type: AWS::EC2::VPC
  Properties:
  CidrBlock: 10.0.0.0/16
  EnableDnsSupport: 'true'
  EnableDnsHostnames: 'true'

**Subnet:**
  Type: AWS::EC2::Subnet
  Properties:
  VpcId:
    Ref: VPC
  CidrBlock: 10.0.0.0/24
AvailabilityZone:
  - Fn::Select:
    - 0
    - Fn::GetAZs: ''

SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Notebook SG1
    VpcId:
      Ref: VPC

ExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - sagemaker.amazonaws.com
          Action:
            - sts:AssumeRole
          Path: /
        ManagedPolicyArns:
          - Fn::Sub: arn:${AWS::Partition}:iam::aws:policy/AmazonSageMakerFullAccess

SageMakerNoteBookInstance:
  Type: AWS::SageMaker::NotebookInstance
  Properties:
    InstanceType: ml.t2.medium
    RoleArn:
      Fn::GetAtt:
        - ExecutionRole
        - Arn
    DirectInternetAccess: Disabled
    SubnetId:
      Fn::GetAtt:
        - Subnet
        - SubnetId
    SecurityGroupIds:
      - Fn::GetAtt:
        - SecurityGroup
        - GroupId

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
ExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - sagemaker.amazonaws.com
          Action:
            - sts:AssumeRole
          Path: /
        ManagedPolicyArns:
          - Fn::Sub: arn:${AWS::Partition}:iam::aws:policy/AmazonSageMakerFullAccess

SageMakerNoteBookInstance:
[CT.SAGEMAKER.PR.2] Require Amazon SageMaker notebook instances to be deployed within a custom Amazon VPC

This control checks whether an Amazon SageMaker notebook instance is configured to launch within a custom Amazon VPC.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::SageMaker::NotebookInstance
- **AWS CloudFormation guard rule:** [CT.SAGEMAKER.PR.2 rule specification](p. 1287)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.SAGEMAKER.PR.2 rule specification](p. 1287)
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.SAGEMAKER.PR.2 example templates](p. 1290)

**Explanation**

As a best practice, we recommend that you keep your resources contained inside a VPC whenever possible, to ensure the secure network protection of your infrastructure.

**Remediation for rule failure**

Set SubnetId to the identifier of an Amazon EC2 subnet and set SecurityGroupIds to a list containing one or more EC2 security group identifiers.

The examples that follow show how to implement this remediation.

**Amazon SageMaker Notebook Instance - Example**

Amazon SageMaker notebook instance configured with Amazon VPC connectivity. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "NotebookInstance": {
        "Type": "AWS::SageMaker::NotebookInstance",
        "Properties": {
            "InstanceType": "ml.t2.large",
            "RoleArn": {
                "Fn::GetAtt": [
                    "ExecutionRole",
                    "Arn"
                ]
            },
            "DirectInternetAccess": "Enabled"
        }
    }
}
```
YAML example

NotebookInstance:
  Type: AWS::SageMaker::NotebookInstance
  Properties:
    InstanceType: ml.t2.large
    RoleArn: !GetAtt 'ExecutionRole.Arn'
    SubnetId: !GetAtt 'Subnet.SubnetId'
    SecurityGroupIds:
      - !GetAtt 'SecurityGroup.GroupId'

CT.SAGEMAKER.PR.2 rule specification

# ###################################################################################################################
##       Rule Specification        ##
# ###################################################################################################################
#
# Rule Identifier:
#   sagemaker_notebook_instance_inside_vpc_check
#
# Description:
#   This control checks whether an Amazon SageMaker notebook instance is configured to launch within a custom Amazon VPC.
#
# Reports on:
#   AWS::SageMaker::NotebookInstance
#
# Rule Parameters:
#   None
#
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any SageMaker notebook instance resources
#     Then: SKIP
#   Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a SageMaker notebook instance resource
# And: 'SubnetId' has not been provided or provided as an empty string or non-valid local reference
# And: 'SecurityGroupIds' has not been provided or provided as an empty list or a list that contains empty string values or non-valid local references
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a SageMaker notebook instance resource
# And: 'SubnetId' has been provided as a non-empty string or valid local reference
# And: 'SecurityGroupIds' has not been provided or provided as an empty list or a list that contains empty string values or non-valid local references
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a SageMaker notebook instance resource
# And: 'SubnetId' has not been provided or provided as an empty string or non-valid local reference
# And: 'SecurityGroupIds' have been provided as a non-empty list containing non-empty string values or valid local references
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a SageMaker notebook instance resource
# And: 'SubnetId' has been provided as a non-empty string or valid local reference
# And: 'SecurityGroupIds' have been provided as a list containing one or more non-empty string values or valid local references
# Then: PASS

# Constants
let SAGEMAKER_NOTEBOOK_INSTANCE_TYPE = "AWS::SageMaker::NotebookInstance"
let INPUT_DOCUMENT = this

# Assignments
let sagemaker_notebook_instances = Resources.[ Type == %SAGEMAKER_NOTEBOOK_INSTANCE_TYPE ]

# Primary Rules
rule sagemaker_notebook_instance_inside_vpc_check when is_cfn_template(%INPUT_DOCUMENT)
sagemaker_notebook_instances not empty {
    check(sagemaker_notebook_instances.Properties)
    [CT.SAGEMAKER.PR.2]: Require Amazon SageMaker notebook instances to be deployed within a custom Amazon VPC
    [FIX]: Set 'SubnetId' to the identifier of an Amazon EC2 subnet and set 'SecurityGroupIds' to a list containing one or more EC2 security group identifiers.
}
rule sagemaker_notebook_instance_inside_vpc_check when is_cfn_hook(%INPUT_DOCUMENT, %SAGEMAKER_NOTEBOOK_INSTANCE_TYPE) {
check(%INPUT_DOCUMENT.%SAGEMAKER_NOTEBOOK_INSTANCE_TYPE.resourceProperties)
<<
[CT.SAGEMAKER.PR.2]: Require Amazon SageMaker notebook instances to be deployed
within a custom Amazon VPC
[FIX]: Set 'SubnetId' to the identifier of an Amazon EC2 subnet and set
'SecurityGroupIds' to a list containing one or more EC2 security group identifiers.
>>
}

rule check(sagemaker_notebook_instance) {
  %sagemaker_notebook_instance {
    # Scenario 2
    SubnetId exists

    # Scenario 3, 4 and 5
    check_is_string_and_not_empty(SubnetId) or
    check_local_references(%INPUT_DOCUMENT, SubnetId, "AWS::EC2::Subnet")

    SecurityGroupIds exists
    SecurityGroupIds is_list
    SecurityGroupIds not empty

    SecurityGroupIds[*] {
      check_is_string_and_not_empty(this) or
      check_local_references(%INPUT_DOCUMENT, this, "AWS::EC2::SecurityGroup")
    }
  }
}

# Utility Rules
#
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}

rule check_is_string_and_not_empty(value) {
  %value {
    this is_string
    this != /\A\s\*/z/
  }
}

rule check_local_references(doc, reference_properties, referenced_resource_type) {
  %reference_properties {
    'Fn::GetAtt' {
      query_for_resource(%doc, this[0], %referenced_resource_type)
      <Local Stack reference was invalid>
    } or Ref {
      query_for_resource(%doc, this, %referenced_resource_type)
      <Local Stack reference was invalid>
    }
  }
}

rule query_for_resource(doc, resource_key, referenced_resource_type) {
  let referenced_resource = %doc.Resources[ keys == %resource_key ]
  %referenced_resource not empty
  %referenced_resource {
CT.SAGEMAKER.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```yaml
Resources:
  NotebookInstance:
    Type: AWS::SageMaker::NotebookInstance
    Properties:
      InstanceType: ml.t2.large
      RoleArn: example-role-arn
      SubnetId: example-subnet-id
      SecurityGroupIds:
        - example-sg-id
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```yaml
Resources:
  NotebookInstance:
    Type: AWS::SageMaker::NotebookInstance
    Properties:
      InstanceType: ml.t2.large
      RoleArn: example-role-arn
```

[CT.SAGEMAKER.PR.3] Require Amazon SageMaker notebook instances to have root access disallowed

This control checks whether Amazon SageMaker notebook instances allow root access.

- **Control objective**: Enforce least privilege
- **Implementation**: AWS CloudFormation guard rule
- **Control behavior**: Proactive
- **Resource types**: AWS::SageMaker::NotebookInstance
- **AWS CloudFormation guard rule**: [CT.SAGEMAKER.PR.3 rule specification (p. 1291)](#)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.SAGEMAKER.PR.3 rule specification (p. 1291)](#)
- For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.SAGEMAKER.PR.3 example templates (p. 1293)](#)

Explanation
By default, when you create a notebook instance, users that log into that notebook instance have root access. Because users with root access have administrator privileges, users have access to edit all files on a notebook instance with root access enabled. In adherence to the principle of least privilege, for security reasons, we recommend that you restrict root access to instance resources whenever possible, to avoid unintentional over-provisioning of permissions.

**Usage considerations**

- Lifecycle configurations associated with an Amazon SageMaker notebook instance always run with root access, even if you turn off root access for users.

**Remediation for rule failure**

Set RootAccess to Disabled.

The examples that follow show how to implement this remediation.

**Amazon SageMaker Notebook Instance - Example**

Amazon SageMaker notebook instance configured with root access turned off. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "NotebookInstance": {
      "Type": "AWS::SageMaker::NotebookInstance",
      "Properties": {
         "InstanceType": "ml.t2.large",
         "RoleArn": {
            "Fn::GetAtt": [
               "ExecutionRole",
               "Arn"
            ],
            "RootAccess": "Disabled"
         }
      }
   }
}
```

**YAML example**

```yaml
NotebookInstance:
   Type: AWS::SageMaker::NotebookInstance
   Properties:
      InstanceType: ml.t2.large
      RoleArn: !GetAtt 'ExecutionRole.Arn'
      RootAccess: Disabled
```

**CT.SAGEMAKER.PR.3 rule specification**

```
# ###################################
##       Rule Specification        ##
#####################################
```
# Rule Identifier:
# sagemaker_notebook_instance_root_access_check
#
# Description:
# This control checks whether Amazon SageMaker notebook instances allow root access.
#
# Reports on:
# AWS::SageMaker::NotebookInstance
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any SageMaker notebook instance resources
#     Then: SKIP
#
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a SageMaker notebook instance resource
#     And: 'RootAccess' has not been provided
#     Then: FAIL
#
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a SageMaker notebook instance resource
#     And: 'RootAccess' has been provided and is set to a value other than 'Disabled'
#     Then: FAIL
#
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains a SageMaker notebook instance resource
#     And: 'RootAccess' has been provided and is set to 'Disabled'
#     Then: PASS
#
# Constants
#
let SAGEMAKER_NOTEBOOK_INSTANCE_TYPE = "AWS::SageMaker::NotebookInstance"
let INPUT_DOCUMENT = this
#
# Assignments
#
let sagemaker_notebook_instances = Resources.*[ Type == %SAGEMAKER_NOTEBOOK_INSTANCE_TYPE ]
#
# Primary Rules
#
rule sagemaker_notebook_instance_root_access_check when is_cfn_template(%INPUT_DOCUMENT)
    %sagemaker_notebook_instances not empty {
        check(%sagemaker_notebook_instances.Properties)
        <<
            [CT.SAGEMAKER.PR.3]: Require Amazon SageMaker notebook instances to have root
            access disallowed
            [FIX]: Set 'RootAccess' to 'Disabled'.
        >>
    }
rule sagemaker_notebook_instance_root_access_check when is_cfn_hook(%INPUT_DOCUMENT, %SAGEMAKER_NOTEBOOK_INSTANCE_TYPE) {
    check(%INPUT_DOCUMENT.%SAGEMAKER_NOTEBOOK_INSTANCE_TYPE.resourceProperties)
    <<
    [CT.SAGEMAKER.PR.3]: Require Amazon SageMaker notebook instances to have root access disallowed
    
    [FIX]: Set 'RootAccess' to 'Disabled'.
    >>
}

rule check(sagemaker_notebook_instance) {
    %sagemaker_notebook_instance {
        # Scenarios 2, 3 and 4
        RootAccess exists
        RootAccess == "Disabled"
    }
}

# Utility Rules
#
rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.SAGEMAKER.PR.3 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
NotebookInstance:
    Type: AWS::SageMaker::NotebookInstance
    Properties:
        InstanceType: ml.t2.large
        RoleArn: example-role-arn
        RootAccess: Disabled

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
NotebookInstance:
    Type: AWS::SageMaker::NotebookInstance
    Properties:
        InstanceType: ml.t2.large
        RoleArn: example-role-arn
        RootAccess: Enabled
Amazon Simple Queue Service (Amazon SQS) controls

Topics
- [CT.SQS.PR.1] Require any Amazon SQS queue to have a dead-letter queue configured (p. 1294)
- [CT.SQS.PR.2] Require any Amazon SQS queue to have encryption at rest configured (p. 1298)

[CT.SQS.PR.1] Require any Amazon SQS queue to have a dead-letter queue configured

This control checks whether an Amazon SQS queue is configured with a dead-letter queue.

- **Control objective:** Improve resiliency
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::SQS::Queue
- **AWS CloudFormation guard rule:** [CT.SQS.PR.1 rule specification (p. 1295)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.SQS.PR.1 rule specification (p. 1295)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.SQS.PR.1 example templates (p. 1298)]

Explanation

The main task of a dead-letter queue is to handle the lifecycle of unconsumed messages. A dead-letter queue lets you set aside and isolate messages that can't be processed correctly, so you can determine why their processing didn't succeed.

**Usage considerations**

- This control applies only to Amazon SQS queues that are not configured as a dead-letter queue with a RedriveAllowPolicy property.

Remediation for rule failure

Create a RedrivePolicy with a deadLetterTargetArn value that's set to the ARN of an Amazon SQS dead-letter queue. For Amazon SQS dead-letter queues, instead provide a redrive configuration in the RedriveAllowPolicy property.

The examples that follow show how to implement this remediation.

Amazon SQS Queue - Example

Amazon SQS queue configured to send messages to a dead-letter queue, if the messages can't be processed (consumed) successfully. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "SQSQueue": {
        "Type": "AWS::SQS::Queue",
```
"Properties": {
  "RedrivePolicy": {
    "deadLetterTargetArn": {
      "Fn::GetAtt": [
        "DLQQueue",
        "Arn"
      ],
      "maxReceiveCount": 3
    }
  }
}

YAML example

```yaml
SQSQueue:
  Type: AWS::SQS::Queue
  Properties:
    RedrivePolicy:
      deadLetterTargetArn: !GetAtt 'DLQQueue.Arn'
      maxReceiveCount: 3
```

CT.SQS.PR.1 rule specification

```plaintext
# #############################################################################
##       Rule Specification       ##
# #############################################################################
#
# Rule Identifier:
#    sqs_dlq_check
#
# Description:
#    This control checks whether an Amazon SQS queue is configured with a dead-letter queue.
#
# Reports on:
#    AWS::SQS::Queue
#
# Evaluates:
#    AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
#    None
#
# Scenarios:
#    Scenario: 1
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#      And: The input document does not contain any SQS queue resources
#      Then: SKIP
#    Scenario: 2
#      Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#      And: The input document contains an SQS queue resource
#      And: 'RedriveAllowPolicy' has been provided on the SQS queue
#      And: 'RedriveAllowPolicy.redrivePermission' is set to 'allowAll' or 'byQueue'
#      Then: SKIP
#    Scenario: 3
```
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an SQS queue resource
# And: 'RedriveAllowPolicy' has not been provided on the SQS queue or 'RedriveAllowPolicy.redrivePermission' has been provided and is set to a value other than 'allowAll' or 'byQueue'
# And: 'RedrivePolicy' has not been provided
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an SQS queue resource
# And: 'RedriveAllowPolicy' has not been provided on the SQS queue or 'RedriveAllowPolicy.redrivePermission' has been provided and is set to a value other than 'allowAll' or 'byQueue'
# And: 'RedrivePolicy' has been provided
# And: 'RedrivePolicy.deadLetterTargetArn' has not been provided or has been provided as an empty string or invalid local reference to an SQS queue
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an SQS queue resource
# And: 'RedriveAllowPolicy' has not been provided on the SQS queue or 'RedriveAllowPolicy.redrivePermission' has been provided and is set to a value other than 'allowAll' or 'byQueue'
# And: 'RedrivePolicy' has been provided
# And: 'RedrivePolicy.deadLetterTargetArn' has been provided as a non-empty string or valid local reference to an SQS queue
# Then: PASS

# Constants

let SQS_QUEUE_TYPE = "AWS::SQS::Queue"
let INPUT_DOCUMENT = this
let DLQ_REDRIVE_PERMISSION = ["allowAll", "byQueue"]

# Assignments

let sqs_queues = Resources.*[ Type == %SQS_QUEUE_TYPE ]

# Primary Rules

rule sqs_dlq_check when is_cfn_template(%INPUT_DOCUMENT)
%{sqs_queues not empty}
  check(%sqs_queues.Properties)
  <<
  [CT.SQS.PR.1]: Require any Amazon SQS queue to have a dead-letter queue configured
  [FIX]: Create a 'RedrivePolicy' with a 'deadLetterTargetArn' value that's set to the ARN of an Amazon SQS dead-letter queue. For Amazon SQS dead-letter queues, instead provide a redrive configuration in the 'RedriveAllowPolicy' property.
  >>
}

rule sqs_dlq_check when is_cfn_hook(%INPUT_DOCUMENT, %SQS_QUEUE_TYPE) {
  check(%INPUT_DOCUMENT.%SQS_QUEUE_TYPE.resourceProperties)
  <<
  [CT.SQS.PR.1]: Require any Amazon SQS queue to have a dead-letter queue configured
  [FIX]: Create a 'RedrivePolicy' with a 'deadLetterTargetArn' value that's set to the ARN of an Amazon SQS dead-letter queue. For Amazon SQS dead-letter queues, instead provide a redrive configuration in the 'RedriveAllowPolicy' property.
  >>
}
# Parameterized Rules

## Scenario 2
RedriveAllowPolicy not exists or filter_is_not_dlq(this)

## Scenario 3
RedrivePolicy exists
RedrivePolicy is_struct

## Scenario 4
RedrivePolicy {
  deadLetterTargetArn exists
  check_is_string_and_not_empty(deadLetterTargetArn) or check_local_references(%INPUT_DOCUMENT, deadLetterTargetArn, %SQS_QUEUE_TYPE)
}

## Utility Rules

## Scenario 2
RedriveAllowPolicy exists
RedriveAllowPolicy is_struct

RedriveAllowPolicy {
  redrivePermission exists
  redrivePermission not in %DLQ_REDRIVE_PERMISSION
}

## Scenario 3
RedrivePolicy exists
RedrivePolicy is_struct

RedrivePolicy {
  redrivePermission exists
  redrivePermission not in %DLQ_REDRIVE_PERMISSION
}

## Scenario 4
RedrivePolicy {
  deadLetterTargetArn exists
  check_is_string_and_not_empty(deadLetterTargetArn) or check_local_references(%INPUT_DOCUMENT, deadLetterTargetArn, %SQS_QUEUE_TYPE)
}

## Utility Rules

## Scenario 2
RedriveAllowPolicy exists
RedriveAllowPolicy is_struct

RedriveAllowPolicy {
  redrivePermission exists
  redrivePermission not in %DLQ_REDRIVE_PERMISSION
}

## Scenario 3
RedrivePolicy exists
RedrivePolicy is_struct

RedrivePolicy {
  redrivePermission exists
  redrivePermission not in %DLQ_REDRIVE_PERMISSION
}

## Scenario 4
RedrivePolicy {
  deadLetterTargetArn exists
  check_is_string_and_not_empty(deadLetterTargetArn) or check_local_references(%INPUT_DOCUMENT, deadLetterTargetArn, %SQS_QUEUE_TYPE)
}

## Utility Rules

## Scenario 2
RedriveAllowPolicy exists
RedriveAllowPolicy is_struct

RedriveAllowPolicy {
  redrivePermission exists
  redrivePermission not in %DLQ_REDRIVE_PERMISSION
}

## Scenario 3
RedrivePolicy exists
RedrivePolicy is_struct

RedrivePolicy {
  redrivePermission exists
  redrivePermission not in %DLQ_REDRIVE_PERMISSION
}

## Scenario 4
RedrivePolicy {
  deadLetterTargetArn exists
  check_is_string_and_not_empty(deadLetterTargetArn) or check_local_references(%INPUT_DOCUMENT, deadLetterTargetArn, %SQS_QUEUE_TYPE)
}

## Utility Rules

## Scenario 2
RedriveAllowPolicy exists
RedriveAllowPolicy is_struct

RedriveAllowPolicy {
  redrivePermission exists
  redrivePermission not in %DLQ_REDRIVE_PERMISSION
}

## Scenario 3
RedrivePolicy exists
RedrivePolicy is_struct

RedrivePolicy {
  redrivePermission exists
  redrivePermission not in %DLQ_REDRIVE_PERMISSION
}

## Scenario 4
RedrivePolicy {
  deadLetterTargetArn exists
  check_is_string_and_not_empty(deadLetterTargetArn) or check_local_references(%INPUT_DOCUMENT, deadLetterTargetArn, %SQS_QUEUE_TYPE)
}

## Utility Rules

## Scenario 2
RedriveAllowPolicy exists
RedriveAllowPolicy is_struct

RedriveAllowPolicy {
  redrivePermission exists
  redrivePermission not in %DLQ_REDRIVE_PERMISSION
}

## Scenario 3
RedrivePolicy exists
RedrivePolicy is_struct

RedrivePolicy {
  redrivePermission exists
  redrivePermission not in %DLQ_REDRIVE_PERMISSION
}

## Scenario 4
RedrivePolicy {
  deadLetterTargetArn exists
  check_is_string_and_not_empty(deadLetterTargetArn) or check_local_references(%INPUT_DOCUMENT, deadLetterTargetArn, %SQS_QUEUE_TYPE)
}

## Utility Rules

## Scenario 2
RedriveAllowPolicy exists
RedriveAllowPolicy is_struct

RedriveAllowPolicy {
  redrivePermission exists
  redrivePermission not in %DLQ_REDRIVE_PERMISSION
}

## Scenario 3
RedrivePolicy exists
RedrivePolicy is_struct

RedrivePolicy {
  redrivePermission exists
  redrivePermission not in %DLQ_REDRIVE_PERMISSION
}

## Scenario 4
RedrivePolicy {
  deadLetterTargetArn exists
  check_is_string_and_not_empty(deadLetterTargetArn) or check_local_references(%INPUT_DOCUMENT, deadLetterTargetArn, %SQS_QUEUE_TYPE)
}
CT.SQS.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  SQSQueue:
    Type: AWS::SQS::Queue
    Properties:
      RedrivePolicy:
        deadLetterTargetArn:
          Fn::GetAtt: [DLQQueue, Arn]
        maxReceiveCount: 3
  DLQQueue:
    Type: AWS::SQS::Queue
    Properties:
      RedriveAllowPolicy:
        redrivePermission: allowAll
```

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
Resources:
  SQSQueue:
    Type: AWS::SQS::Queue
    Properties: {}
```

[CT.SQS.PR.2] Require any Amazon SQS queue to have encryption at rest configured

This control checks whether an Amazon SQS queue is encrypted at rest.

- **Control objective**: Encrypt data at rest
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::SQS::Queue
- **AWS CloudFormation guard rule**: CT.SQS.PR.2 rule specification (p. 1300)

Details and examples
• For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.SQS.PR.2 rule specification](p. 1300)

• For examples of PASS and FAIL AWS CloudFormation Templates related to this control, see: [CT.SQS.PR.2 example templates](p. 1302)

**Explanation**

Server-side encryption (SSE) allows you to transmit sensitive data in encrypted queues. To protect the content of messages in queues, SSE uses KMS keys.

**Remediation for rule failure**

Set `SqsManagedSseEnabled` to `true` or set an AWS KMS key identifier in the `KmsMasterKeyId` property.

The examples that follow show how to implement this remediation.

**Amazon SQS Queue - Example One**

Amazon SQS queue configured to encrypt data at rest with server-side encryption enabled, by means of SQS managed encryption keys (SSE-SQS). The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "SQSQueue": {
    "Type": "AWS::SQS::Queue",
    "Properties": {
      "SqsManagedSseEnabled": true
    }
  }
}
```

**YAML example**

```yaml
SQSQueue:
  Type: AWS::SQS::Queue
  Properties:
    SqsManagedSseEnabled: true
```

The examples that follow show how to implement this remediation.

**Amazon SQS Queue - Example Two**

Amazon SQS queue configured to encrypt data at rest with server-side encryption enabled, by means of AWS KMS (SSE-KMS). The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "SQSQueue": {
    "Type": "AWS::SQS::Queue",
    "Properties": {
      "SqsManagedSseEnabled": true
    }
  }
}
```
"KmsMasterKeyId": {  
  "Ref": "KMSKey"
}
}
}

YAML example

SQSQueue:
  Type: AWS::SQS::Queue
  Properties:
    KmsMasterKeyId: !Ref 'KMSKey'

CT.SQS.PR.2 rule specification

# #####################################################################
##       Rule Specification        
# #####################################################################
# # Rule Identifier: # sqs_queue_encrypted_check  
# # Description: # This control checks whether an Amazon SQS queue is encrypted at rest.  
# # Reports on: # AWS::SQS::Queue  
# # Evaluates: # AWS CloudFormation, AWS CloudFormation hook  
# # Rule Parameters: None  
# # Scenarios:  
# # Scenario: 1 # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document  
#                        And: The input document does not contain any SQS queue resources  
#                        Then: SKIP  
# # Scenario: 2 # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document  
#                        And: The input document contains an SQS queue resource  
#                        And: 'KmsMasterKeyId' or 'SqsManagedSseEnabled' have not been provided on the SQS queue resource  
#                        Then: FAIL  
# # Scenario: 3 # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document  
#                        And: The input document contains an SQS queue resource  
#                        And: 'KmsMasterKeyId' has not been provided  
#                        And: 'SqsManagedSseEnabled' has been provided and set to bool(false)  
#                        Then: FAIL  
# # Scenario: 4 # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an SQS queue resource
# And: 'KmsMasterKeyId' has been provided as an empty string or invalid local
# reference to a KMS key or alias
# Then: FAIL
# Scenario: 5
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an SQS queue resource
# And: 'SqsManagedSseEnabled' is not provided or set to bool(false)
# And: 'KmsMasterKeyId' is provided as a non-empty string or local reference to a KMS
# key
# Then: PASS
# Scenario: 6
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains an SQS queue resource
# And: 'KmsMasterKeyId' is not provided
# And: 'SqsManagedSseEnabled' is provided and set to bool(true)
# Then: PASS

# Constants
#
let SQS_QUEUE_TYPE = "AWS::SQS::Queue"
let INPUT_DOCUMENT = this
#
# Assignments
#
let sqs_queues = Resources. *[ Type == %SQS_QUEUE_TYPE ]
#
# Primary Rules
#
rule sqs_queue_encrypted_check when is_cfn_template(%INPUT_DOCUMENT) {
    check(%sqs_queues.Properties) %sqs_queues not empty {
        [CT.SQS.PR.2]: Require any Amazon SQS queue to have encryption at rest configured
        [FIX]: Set 'SqsManagedSseEnabled' to 'true' or set an AWS KMS key identifier in the
        'KmsMasterKeyId' property.
        >>
    }
}
rule sqs_queue_encrypted_check when is_cfn_hook(%INPUT_DOCUMENT, %SQS_QUEUE_TYPE) {
    check(%INPUT_DOCUMENT.%SQS_QUEUE_TYPE.resourceProperties) %sqs_queues not empty {
        [CT.SQS.PR.2]: Require any Amazon SQS queue to have encryption at rest configured
        [FIX]: Set 'SqsManagedSseEnabled' to 'true' or set an AWS KMS key identifier in the
        'KmsMasterKeyId' property.
        >>
    }
}
#
# Parameterized Rules
#
rule check(sqs_queue) {
    %sqs_queue {
        check_sse_enabled(this) or
        check_kms_valid(this)
    }
}
rule check_sse_enabled(sqs_queue) {
    # Scenario 2
    SqsManagedSseEnabled exists
}
# Scenario 3, 6
KmsMasterKeyId not exists
SqsManagedSseEnabled == true
}

rule check_kms_valid(sqs_queue) {
# Scenario 2
KmsMasterKeyId exists

# Scenario 4, 5
ccheck_is_string_and_not_empty(KmsMasterKeyId) or
ccheck_local_references(%INPUT_DOCUMENT, KmsMasterKeyId, "AWS::KMS::Key") or
ccheck_local_references(%INPUT_DOCUMENT, KmsMasterKeyId, "AWS::KMS::Alias")

# Scenario 5
SqsManagedSseEnabled not exists or
SqsManagedSseEnabled == false
}

# Utility Rules
#
rule is_cfn_template(doc) {
%doc {
  AWSTemplateFormatVersion exists or
  Resources exists
}
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
%doc.%RESOURCE_TYPE.resourceProperties exists
}

rule check_is_string_and_not_empty(value) {
%value {
  this is_string
  this != /\A\s*\z/
}
}

rule check_local_references(doc, reference_properties, referenced_resource_type) {
%reference_properties {
  "Fn::GetAtt" {
    query_for_resource(%doc, this[0], %referenced_resource_type)
    <<Local Stack reference was invalid>>
  } or Ref {
    query_for_resource(%doc, this, %referenced_resource_type)
    <<Local Stack reference was invalid>>
  }
}

rule query_for_resource(doc, resource_key, referenced_resource_type) {
let referenced_resource = %doc.Resources[ keys == %resource_key ]
%referenced_resource not empty
%referenced_resource {
  Type == %referenced_resource_type
}
}

CT.SQS.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.
PASS Example - Use this template to verify a compliant resource creation.

Resources:
KMSKey:
  Type: AWS::KMS::Key
  Properties:
    KeyPolicy:
      Version: 2012-10-17
      Id: key-default-1
      Statement:
        - Sid: Enable IAM User Permissions
          Effect: Allow
          Principal:
            AWS: '*'
          Action: 'kms:*
          Resource: '*'
SQSQueue:
  Type: AWS::SQS::Queue
  Properties:
    KmsMasterKeyId:
      Ref: KMSKey

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
SQSQueue:
  Type: AWS::SQS::Queue
  Properties: {}

AWS Step Functions controls

Topics
- [CT.STEPFUNCTIONS.PR.1] Require an AWS Step Functions state machine to have logging activated (p. 1303)
- [CT.STEPFUNCTIONS.PR.2] Require an AWS Step Functions state machine to have AWS X-Ray tracing activated (p. 1311)

[CT.STEPFUNCTIONS.PR.1] Require an AWS Step Functions state machine to have logging activated

This control checks whether an AWS Step Functions state machine has logging enabled.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::StepFunctions::StateMachine
- **AWS CloudFormation guard rule:** CT.STEPFUNCTIONS.PR.1 rule specification (p. 1305)

Details and examples
For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.STEPFUNCTIONS.PR.1 rule specification (p. 1305)]

For examples of PASS and FAIL CloudFormation templates related to this control, see: [CT.STEPFUNCTIONS.PR.1 example templates (p. 1308)]

Explanation

Defining a logging configuration for your state machines allows you to track their execution history and results. This configuration allows you to track failed events that occur on a state machine, and this insight into errors can assist you when you’re troubleshooting issues.

Remediation for rule failure

In `LoggingConfiguration`, set `Level` to `ERROR` or `ALL`, and set `Destinations` to a list with one or more valid Amazon CloudWatch Logs log group ARNs.

The examples that follow show how to implement this remediation.

**AWS Step Functions State Machine - Example**

AWS Step Functions state machine configured to send logs to Amazon CloudWatch Logs. The example is shown in JSON and in YAML.

**JSON example**

```json
{
  "StateMachine": {
    "Type": "AWS::StepFunctions::StateMachine",
    "Properties": {
      "StateMachineType": "STANDARD",
      "DefinitionString": "{"StartAt": "Sample","States": {"Sample": {"Type": "Task", "Resource":"arn:aws:lambda:us-east-1:111122223333:function:SampleFunction", "End": true}}}}",
      "RoleArn": {
        "Fn::GetAtt": [
          "StepFunctionExecutionRole",
          "Arn"
        ],
      },
      "LoggingConfiguration": {
        "Level": "ALL",
        "Destinations": [
          {
            "CloudWatchLogsLogGroup": {
              "LogGroupArn": {
                "Fn::GetAtt": [
                  "LogGroup",
                  "Arn"
                ]
              }
            }
          }
        ]
      }
    }
  }
}
```

**YAML example**

```yaml
...
```
StateMachine:
  Type: AWS::StepFunctions::StateMachine
  Properties:
    StateMachineType: STANDARD
    DefinitionString: '{"StartAt": "Sample","States": [{"Sample": {"Type": "Task",
       "End": true}}],
    RoleArn: !GetAtt 'StepFunctionExecutionRole.Arn'
    LoggingConfiguration:
      Level: ALL
      Destinations:
        - CloudWatchLogsLogGroup:
            LogGroupArn: !GetAtt 'LogGroup.Arn'

CT.STEPFUNCTIONS.PR.1 rule specification

# #############################################################################
##       Rule Specification        ##
# #############################################################################
#
# Rule Identifier:  
#  step_functions_state_machine_logging_enabled_check
#
# Description: 
#   This control checks whether an AWS Step Functions state machine has logging enabled.
# 
# Reports on: 
#  AWS::StepFunctions::StateMachine
#
# Evaluates: 
#  AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters: 
#   None
#
# Scenarios: 
#  Scenario: 1 
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document does not contain any StepFunctions state machine resources
#       Then: SKIP 
#  Scenario: 2 
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document contains a StepFunctions state machine resource
#       And: 'LoggingConfiguration' has not been provided
#       Then: FAIL 
#  Scenario: 3 
#    Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#       And: The input document contains a StepFunctions state machine resource
#       And: 'LoggingConfiguration' has been provided
#       And: In 'LoggingConfiguration', 'Level' has not been provided or provided and set to a value other than 'ERROR' or 'ALL'
#       And: In 'LoggingConfiguration', 'Destinations' has not been provided or provided as an empty list or list containing empty strings or non-valid local references
#       Then: FAIL 
#  Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a StepFunctions state machine resource
# And: 'LoggingConfiguration' has been provided
# And: In 'LoggingConfiguration', 'Level' has not been provided or provided and set to a value other than 'ERROR' or 'ALL'
# And: In 'LoggingConfiguration', 'Destinations' has not been provided or provided as an empty list or list containing empty strings or non-valid local references
# Then: FAIL

Scenario: 5

# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a StepFunctions state machine resource
# And: 'LoggingConfiguration' has been provided
# And: In 'LoggingConfiguration', 'Level' has not been provided or provided and set to a value other than 'ERROR' or 'ALL'
# And: In 'LoggingConfiguration', 'Destinations' has been provided as a list containing non-empty strings or valid local references
# Then: FAIL

Scenario: 6

# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a StepFunctions state machine resource
# And: 'LoggingConfiguration' has been provided
# And: In 'LoggingConfiguration', 'Level' has been provided and set to 'ERROR' or 'ALL'
# And: In 'LoggingConfiguration', 'Destinations' has been provided as a list containing non-empty strings or valid local references
# Then: PASS

# Constants

let STEP_FUNCTIONS_STATE_MACHINE_TYPE = "AWS::StepFunctions::StateMachine"
let ALLOWED_LOGGING_LEVELS = [ "ERROR", "ALL" ]
let INPUT_DOCUMENT = this

# Assignments

let step_functions_state_machines = Resources.*[ Type == %STEP_FUNCTIONS_STATE_MACHINE_TYPE ]

# Primary Rules

rule step_functions_state_machine_logging_enabled_check when
    is_cfn_template(%INPUT_DOCUMENT)
    %step_functions_state_machines
    not empty {
        check(%step_functions_state_machines.Properties)
        <<
            [CT.STEPFUNCTIONS.PR.1]: Require an AWS Step Functions state machine to have logging activated.
            [FIX]: In 'LoggingConfiguration', set 'Level' to 'ERROR' or 'ALL', and set 'Destinations' to a list with one or more valid Amazon CloudWatch Logs log group ARNs.
        >>
    }

rule step_functions_state_machine_logging_enabled_check when is_cfn_hook(%INPUT_DOCUMENT, %STEP_FUNCTIONS_STATE_MACHINE_TYPE) {
AWS Control Tower User Guide
Proactive controls
check(%INPUT_DOCUMENT.%STEP_FUNCTIONS_STATE_MACHINE_TYPE.resourceProperties)
<<
[CT.STEPFUNCTIONS.PR.1]: Require an AWS Step Functions state machine to have
logging activated
[FIX]: In 'LoggingConfiguration', set 'Level' to 'ERROR' or 'ALL', and set
'Destinations' to a list with one or more valid Amazon CloudWatch Logs log group ARNs.
>>

}

rule check(step_functions_state_machine) {
%step_functions_state_machine {
# Scenario 2
LoggingConfiguration exists
LoggingConfiguration is_struct
LoggingConfiguration {
# Scenarios 3, 4, 5 and 6
Level exists
Level in %ALLOWED_LOGGING_LEVELS
Destinations exists
Destinations is_list
Destinations not empty
Destinations[*] {
CloudWatchLogsLogGroup exists
CloudWatchLogsLogGroup is_struct
CloudWatchLogsLogGroup {
LogGroupArn exists
check_is_string_and_not_empty(LogGroupArn) or
check_local_references(%INPUT_DOCUMENT, LogGroupArn,
"AWS::Logs::LogGroup")
}
}
}
}

}

#
# Utility Rules
#
rule is_cfn_template(doc) {
%doc {
AWSTemplateFormatVersion exists
Resources exists
}
}

or

rule is_cfn_hook(doc, RESOURCE_TYPE) {
%doc.%RESOURCE_TYPE.resourceProperties exists
}
rule check_is_string_and_not_empty(value) {
%value {
this is_string
this != /\A\s*\z/
}
}
rule check_local_references(doc, reference_properties, referenced_resource_type) {
%reference_properties {
'Fn::GetAtt' {
query_for_resource(%doc, this[0], %referenced_resource_type)
<<Local Stack reference was invalid>>

1307


CT.STEPFUNCTIONS.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

```
Resources:
  StepFunctionExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - states.amazonaws.com
            Action:
              - sts:AssumeRole
            Condition:
              ArnLike:
                aws:SourceArn:
                Fn::Sub: arn:${AWS::Partition}:states:${AWS::Region}:${AWS::AccountId}:*
              StringEquals:
                aws:SourceAccount:
                Ref: AWS::AccountId
            Path: /
            Policies:
              - PolicyName: StepFunctionLoggingPolicy
                PolicyDocument:
                  Version: '2012-10-17'
                  Statement:
                    - Effect: Allow
                      Action:
                        - logs:CreateLogDelivery
                        - logs:GetLogDelivery
                        - logs:UpdateLogDelivery
                        - logs:DeleteLogDelivery
                        - logs:ListLogDeliveries
                        - logs:PutLogEvents
                        - logs:PutResourcePolicy
                        - logs:DescribeResourcePolicies
                        - logs:DescribeLogGroups
                      Resource: '*'
              
LogGroup:
    Type: AWS::Logs::LogGroup
```
Properties: {}
StateMachine:
  Type: AWS::StepFunctions::StateMachine
Properties:
  StateMachineType: STANDARD
DefinitionString: ' {
  "StartAt": "Example",
  "States": {
    "Example": {
      "Type": "Task",
      "End": true
    }
  }
}'
RoleArn:
  Fn::GetAtt:
    - StepFunctionExecutionRole
    - Arn
LoggingConfiguration:
  Level: ALL
Destinations:
  - CloudWatchLogsLogGroup:
    LogGroupArn:
      Fn::GetAtt:
        - LogGroup
        - Arn

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  StepFunctionExecutionRole:
    Type: AWS::IAM::Role
Properties:
  AssumeRolePolicyDocument:
    Version: '2012-10-17'
    Statement:
      - Effect: Allow
        Principal:
          Service:
            - states.amazonaws.com
        Action:
          - sts:AssumeRole
        Condition:
          ArnLike:
            aws:SourceArn:
              Fn::Sub: arn:${AWS::Partition}:states:${AWS::Region}:${AWS::AccountId}:
            StringEquals:
              aws:SourceAccount:
                Ref: AWS::AccountId
        Path: /
    Policies:
      - PolicyName: StepFunctionLoggingPolicy
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - logs:CreateLogDelivery
                - logs:GetLogDelivery
                - logs:UpdateLogDelivery
                - logs:DeleteLogDelivery
                - logs:ListLogDeliveries
                - logs:PutLogEvents
                - logs:PutResourcePolicy
                - logs:DescribeResourcePolicies
                - logs:DescribeLogGroups
                Resource: '*'
**AWS Control Tower User Guide**

**Proactive controls**

```
Type: AWS::StepFunctions::StateMachine
Properties:
  StateMachineType: STANDARD
  DefinitionString: '{"StartAt": "Example","States": [{"Example": {"Type": "Task", "Resource": "arn:aws:lambda:us-east-1:111122223333:function:ExampleFunction", "End": true}}]}'
  RoleArn:
    Fn::GetAtt:
    - StepFunctionExecutionRole
    - Arn

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  StepFunctionExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
        - Effect: Allow
          Principal:
            Service:
              - states.amazonaws.com
          Action:
            - sts:AssumeRole
          Condition:
            ArnLike:
              aws:SourceArn:
                Fn::Sub: arn:${AWS::Partition}:states:${AWS::Region}:${AWS::AccountId}::*
              StringEquals:
                aws:SourceAccount:
                  Ref: AWS::AccountId
          Path: /
      Policies:
        - PolicyName: StepFunctionLoggingPolicy
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
            - Effect: Allow
              Action:
                - logs:CreateLogDelivery
                - logs:GetLogDelivery
                - logs:UpdateLogDelivery
                - logs:DeleteLogDelivery
                - logs:ListLogDeliveries
                - logs:PutLogEvents
                - logs:PutResourcePolicy
                - logs:DescribeResourcePolicies
                - logs:DescribeLogGroups
              Resource: '*'
  StateMachine:
    Type: AWS::StepFunctions::StateMachine
    Properties:
      StateMachineType: STANDARD
      DefinitionString: '{"StartAt": "Example","States": [{"Example": {"Type": "Task", "Resource": "arn:aws:lambda:us-east-1:111122223333:function:ExampleFunction", "End": true}}]}'
      RoleArn:
        Fn::GetAtt:
        - StepFunctionExecutionRole
        - Arn
```
Proactive controls

LoggingConfiguration:
Level: 'OFF'

[CT.STEPFUNCTIONS.PR.2] Require an AWS Step Functions state machine to have AWS X-Ray tracing activated

This control checks whether an AWS Step Functions state machine has AWS X-Ray tracing enabled.

- **Control objective:** Establish logging and monitoring
- **Implementation:** AWS CloudFormation guard rule
- **Control behavior:** Proactive
- **Resource types:** AWS::StepFunctions::StateMachine
- **AWS CloudFormation guard rule:** CT.STEPFUNCTIONS.PR.2 rule specification (p. 1312)

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: CT.STEPFUNCTIONS.PR.2 rule specification (p. 1312)
- For examples of PASS and FAIL CloudFormation templates related to this control, see: CT.STEPFUNCTIONS.PR.2 example templates (p. 1314)

Explanation

A tracing configuration allows your state machine to send tracing data to AWS X-Ray, so you can visualize the components of your state machine, identify performance bottlenecks, and troubleshoot requests that resulted in errors.

Remediation for rule failure

In the TracingConfiguration property, set the value of Enabled to true.

The examples that follow show how to implement this remediation.

**AWS Step Functions State Machine - Example**

An AWS Step Functions state machine configured to send trace data to AWS X-Ray. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "StateMachine": {
        "Type": "AWS::StepFunctions::StateMachine",
        "Properties": {
            "StateMachineType": "STANDARD",
            "DefinitionString": "\"StartAt\": \"Sample\", \"States\": [{\"Sample\": {\"Type \": \"Task\", \"Resource\": \"arn:aws:lambda:us-east-1:111122223333:function:SampleFunction \", \"End\": true}}],
            "RoleArn": {
                "Fn::GetAtt": [
                    "StepFunctionExecutionRole",
                    "Arn"
                ]
            }
        }
    }
}
```

1311
"LoggingConfiguration": {
    "Level": "ALL",
    "Destinations": [
        "CloudWatchLogsLogGroup": {
            "LogGroupArn": {
                "Fn::GetAtt": [
                    "LogGroup",
                    "Arn"
                ]
            }
        }
    ]
},
"TracingConfiguration": {
    "Enabled": true
}
}

YAML example

StateMachine:
  Type: AWS::StepFunctions::StateMachine
Properties:
  StateMachineType: STANDARD
  DefinitionString: '{"StartAt": "Sample","States": {"Sample": {"Type": "Task",
    "End": true}},
RoleArn: !GetAtt 'StepFunctionExecutionRole.Arn'
LoggingConfiguration:
  Level: ALL
  Destinations:
    - CloudWatchLogsLogGroup:
      LogGroupArn: !GetAtt 'LogGroup.Arn'
TracingConfiguration:
  Enabled: true

CT.STEPFUNCTIONS.PR.2 rule specification

# #############################################################################
##       Rule Specification       ##
# #############################################################################

# Rule Identifier:
#  step_functions_state_machine_tracing_enabled_check
# Description:
#  This control checks whether an AWS Step Functions state machine has AWS X-Ray tracing enabled.
# Reports on:
#  AWS::StepFunctions::StateMachine
# Evaluates:
#  AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document does not contain any StepFunctions state machine resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a StepFunctions state machine resource
# And: 'TracingConfiguration' has not been provided
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a StepFunctions state machine resource
# And: 'TracingConfiguration' has been provided
# And: In 'TracingConfiguration', 'Enabled' has not been provided or provided and set
to a value other
# than bool(true)
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
# And: The input document contains a StepFunctions state machine resource
# And: 'TracingConfiguration' has been provided
# And: In 'TracingConfiguration', 'Enabled' has been provided and set to bool(true)
# Then: PASS
#
# Constants
#
let STEP_FUNCTIONS_STATE_MACHINE_TYPE = "AWS::StepFunctions::StateMachine"
let INPUT_DOCUMENT = this
#
# Assignments
#
let step_functions_state_machines = Resources.*[ Type ==
%STEP_FUNCTIONS_STATE_MACHINE_TYPE ]
#
# Primary Rules
#
rule step_functions_state_machine_tracing_enabled_check when
is_cfn_template(%INPUT_DOCUMENT)
%step_functions_state_machines
not empty {
    check(%step_functions_state_machines.Properties)
    <<
    [CT.STEPFUNCTIONS.PR.2]: Require an AWS Step Functions state machine to have AWS X-Ray tracing activated
    [FIX]: In the 'TracingConfiguration' property, set the value of 'Enabled' to true.
    >>
}

rule step_functions_state_machine_tracing_enabled_check when is_cfn_hook(%INPUT_DOCUMENT,
%STEP_FUNCTIONS_STATE_MACHINE_TYPE) {
    check(%INPUT_DOCUMENT.%STEP_FUNCTIONS_STATE_MACHINE_TYPE.resourceProperties)
    <<
    [CT.STEPFUNCTIONS.PR.2]: Require an AWS Step Functions state machine to have AWS X-Ray tracing activated
}
[FIX]: In the 'TracingConfiguration' property, set the value of 'Enabled' to true.

```yaml
rule check(step_functions_state_machine) {
  %step_functions_state_machine {
    # Scenario 2
    TracingConfiguration exists
    TracingConfiguration is_struct
    TracingConfiguration {   # Scenarios 3 and 4
      Enabled exists
      Enabled == true
    }
  }
}
```

# Utility Rules

```yaml
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
```

```yaml
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

**CT.STEPFUNCTIONS.PR.2 example templates**

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```yaml
Resources:
  StepFunctionExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - states.amazonaws.com
            Action:
              - sts:AssumeRole
            Condition:
              ArnLike:
                awss:SourceArn:
                Fn::Sub: arn:${AWS::Partition}:states:${AWS::Region}:${AWS::AccountId}::*
                StringEquals:
                  awss:SourceAccount:
                  Ref: AWS::AccountId
        Path: /
        Policies:
          - PolicyName: StepFunctionLoggingPolicy
```
PolicyDocument:
  Version: '2012-10-17'
  Statement:
    - Effect: Allow
    - Action:
      - logs:CreateLogDelivery
      - logs:GetLogDelivery
      - logs:UpdateLogDelivery
      - logs:DeleteLogDelivery
      - logs:ListLogDeliveries
      - logs:PutLogEvents
      - logs:PutResourcePolicy
      - logs:DescribeResourcePolicies
      - logs:DescribeLogGroups
    Resource: '*'
    - PolicyName: StepFunctionTracingPolicy

PolicyDocument:
  Version: '2012-10-17'
  Statement:
    - Effect: Allow
    - Action:
      - xray:PutTraceSegments
      - xray:PutTelemetryRecords
      - xray:GetSamplingRules
      - xray:GetSamplingTargets
    Resource: '*'

LogGroup:
  Type: AWS::Logs::LogGroup
  Properties: {}

StateMachine:
  Type: AWS::StepFunctions::StateMachine
  Properties:
    StateMachineName:
      Fn::Sub: Example-StateMachine-$(AWS::StackName)
    StateMachineType: STANDARD
    DefinitionString: '{"StartAt": "Example","States": [{"Example": {"Type": "Task",
    "End": true}}]}
    RoleArn:
      Fn::GetAtt:
        - StepFunctionExecutionRole
        - Arn
    LoggingConfiguration:
      Level: ALL
      Destinations:
        - CloudWatchLogsLogGroup:
          LogGroupArn:
            Fn::GetAtt:
              - LogGroup
              - Arn
    TracingConfiguration:
      Enabled: true

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
  StepFunctionExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
- Effect: Allow
  Principal:
    Service:
    - states.amazonaws.com
  Action:
    - sts:AssumeRole
  Condition:
    ArnLike:
      aws:SourceArn:
     Fn::Sub: arn:${AWS::Partition}:states:${AWS::Region}:${AWS::AccountId}::*
    StringEquals:
      aws:SourceAccount:
      Ref: AWS::AccountId
  Path: /
  Policies:
  - PolicyName: StepFunctionLoggingPolicy
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Action:
            - logs:CreateLogDelivery
            - logs:GetLogDelivery
            - logs:UpdateLogDelivery
            - logs:DeleteLogDelivery
            - logs:ListLogDeliveries
            - logs:PutLogEvents
            - logs:PutResourcePolicy
            - logs:DescribeResourcePolicies
            - logs:DescribeLogGroups
          Resource: '*'
  - PolicyName: StepFunctionTracingPolicy
    PolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Action:
            - xray:PutTraceSegments
            - xray:PutTelemetryRecords
            - xray:GetSamplingRules
            - xray:GetSamplingTargets
          Resource: '*'

LogGroup:
  Type: AWS::Logs::LogGroup
  Properties: {}

StateMachine:
  Type: AWS::StepFunctions::StateMachine
  Properties:
    StateMachineName:
      Fn::Sub: Example-StateMachine-{$AWS::StackName}
    StateMachineType: STANDARD
    DefinitionString: '{"StartAt": "Example","States": [{"Example": {"Type": "Task","Resource": "arn:aws:lambda:us-east-1:111122223333:function:ExampleFunction","End": true}}]}
    RoleArn:
      Fn::GetAtt:
        - StepFunctionExecutionRole
        - Arn
    LoggingConfiguration:
      Level: ALL
      Destinations:
        - CloudWatchLogsLogGroup:
          LogGroupArn:
            Fn::GetAtt:
              - LogGroup
              - Arn
TracingConfiguration:
   Enabled: false

AWS WAF regional controls

Topics
- [CT.WAF-REGIONAL.PR.1] Require any AWS WAF regional rule to have a condition (p. 1317)
- [CT.WAF-REGIONAL.PR.2] Require any AWS WAF regional web access control list (ACL) to have a rule or rule group (p. 1320)

[CT.WAF-REGIONAL.PR.1] Require any AWS WAF regional rule to have a condition

This control checks whether an AWS WAF Classic Regional rule contains any conditions.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::WAFRegional::Rule
- **AWS CloudFormation guard rule:** [CT.WAF-REGIONAL.PR.1 rule specification (p. 1318)]

Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: [CT.WAF-REGIONAL.PR.1 rule specification (p. 1318)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.WAF-REGIONAL.PR.1 example templates (p. 1320)]

Explanation

An AWS WAF (web application firewall) Regional rule can contain multiple conditions. The rule count's conditions allow for traffic inspection, based on a defined action, such as allow, block, or count. Without any conditions, the traffic passes without inspection. An AWS WAF Regional rule with no conditions, but with a name or tag suggesting allow, block, or count, could lead to the inaccurate assumption that one of those actions is occurring.

Remediation for rule failure

Provide one or more AWS WAF rule conditions within the Predicates property.

The examples that follow show how to implement this remediation.

**AWS WAF Classic Regional Rule - Example**

AWS WAF Classic Regional rule configured with an IP match predicate. The example is shown in JSON and in YAML.

**JSON example**

```json
{
   "WafRegionalRule": {
```
YAML example

WafRegionalRule:
  Type: AWS::WAFRegional::Rule
  Properties:
    Name: SampleRule
    MetricName: SampleRuleMetric
    Predicates:
      - DataId: !Ref 'IPSet'
        Negated: false
        Type: IPMatch

CT.WAF-REGIONAL.PR.1 rule specification

# #################################################################
## Rule Specification ##
# #################################################################
# Rule Identifier:
#   waf_regional_rule_not_empty_check
# Description:
#   This control checks whether a AWS WAF Classic Regional rule contains any conditions.
# Reports on:
#   AWS::WAFRegional::Rule
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
#   None
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
#     And: The input document does not contain any AWS WAF Classic Regional rule resources
#     Then: SKIP
#   Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an AWS WAF Classic Regional resource
# And: 'Predicates' has not been provided
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an AWS WAF Classic Regional rule resource
# And: 'Predicates' has been provided as an empty list
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains an AWS WAF Classic Regional rule resource
# And: 'Predicates' has been provided as a non-empty list
# Then: PASS

# Constants
#
let WAF_REGIONAL_RULE_TYPE = "AWS::WAFRegional::Rule"
let INPUT_DOCUMENT = this
#
# Assignments
#
let waf_regional_rules = Resources.*[ Type == %WAF_REGIONAL_RULE_TYPE ]
#
# Primary Rules
#
rule waf_regional_rule_not_empty_check when is_cfn_template(%INPUT_DOCUMENT)
%waf_regional_rules not empty {
    check(%waf_regional_rules.Properties)
    <<
        [CT.WAF-REGIONAL.PR.1]: Require any AWS WAF regional rule to have a condition
        [FIX]: Provide one or more AWS WAF rule conditions within the 'Predicates'
        property.
    >>
}
rule waf_regional_rule_not_empty_check when is_cfn_hook(%INPUT_DOCUMENT,
%WAF_REGIONAL_RULE_TYPE) {
    check(%INPUT_DOCUMENT.%WAF_REGIONAL_RULE_TYPE.resourceProperties)
    <<
        [CT.WAF-REGIONAL.PR.1]: Require any AWS WAF regional rule to have a condition
        [FIX]: Provide one or more AWS WAF rule conditions within the 'Predicates'
        property.
    >>
}
#
# Parameterized Rules
#
rule check(waf_regional_rule) {
    %waf_regional_rule {
        # Scenario 2, 3 and 4
        Predicates exists
        Predicates is_list
        Predicates not empty
    }
}
#
# Utility Rules
CT.WAF-REGIONAL.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
- IPSetDenylist:
  - Type: AWS::WAFRegional::IPSet
  - Properties:
    - Name: IPSet for deny listed IP addresses
    - IPSetDescriptors:
      - Type: IPV4
      - Value: 192.0.2.44/32
- WafRegionalRule:
  - Type: AWS::WAFRegional::Rule
  - Properties:
    - Name: ExampleRule
    - MetricName: ExampleRuleMetric
    - Predicates:
      - Type: "IPMatch"
      - DataId:
        - Ref: IPSetDenylist
        - Negated: false

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
- WafRegionalRule:
  - Type: AWS::WAFRegional::Rule
  - Properties:
    - Name: ExampleRule
    - MetricName: ExampleRuleMetric

[CT.WAF-REGIONAL.PR.2] Require any AWS WAF regional web access control list (ACL) to have a rule or rule group

This control checks whether an AWS WAF Classic Regional web ACL contains any WAF rules or rule groups.

- Control objective: Limit network access
AWS Control Tower User Guide
Proactive controls

- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::WAFRegional::WebACL
- **AWS CloudFormation guard rule**: [CT.WAF-REGIONAL.PR.2 rule specification](#)

### Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.WAF-REGIONAL.PR.2 rule specification](#)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.WAF-REGIONAL.PR.2 example templates](#)

### Explanation

An AWS WAF Regional web access control list (ACL) can contain a collection of rules and rule groups that inspect and control web requests. If a web ACL is empty, the web traffic can pass without being detected or acted upon by WAF, depending on the default action.

#### Remediation for rule failure

Provide one or more AWS WAF rules within the `Rules` property.

The examples that follow show how to implement this remediation.

**AWS WAF Classic Regional web ACL - Example**

AWS WAF Classic Regional web ACL configured with a rule to block requests based on an IP set match. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "WafRegionalWebAcl": {
        "Type": "AWS::WAFRegional::WebACL",
        "Properties": {
            "Name": "SampleWebACL",
            "DefaultAction": {
                "Type": "ALLOW"
            },
            "MetricName": "SampleWebACLMetric",
            "Rules": [
                {
                    "Action": {
                        "Type": "BLOCK"
                    },
                    "Priority": 1,
                    "RuleId": {
                        "Ref": "IPSetRule"
                    }
                }
            ]
        }
    }
}
```

**YAML example**

```yaml
{
    "WafRegionalWebAcl": {
        "Type": "AWS::WAFRegional::WebACL",
        "Properties": {
            "Name": "SampleWebACL",
            "DefaultAction": {
                "Type": "ALLOW"
            },
            "MetricName": "SampleWebACLMetric",
            "Rules": [
                {
                    "Action": {
                        "Type": "BLOCK"
                    },
                    "Priority": 1,
                    "RuleId": {
                        "Ref": "IPSetRule"
                    }
                }
            ]
        }
    }
}
```
WafRegionalWebAcl:
  Type: AWS::WAFRegional::WebACL
Properties:
  Name: SampleWebACL
  DefaultAction:
    Type: ALLOW
  MetricName: SampleWebACLMetric
Rules:
  - Action:
      Type: BLOCK
    Priority: 1
    RuleId: !Ref 'IPSetRule'

CT.WAF-REGIONAL.PR.2 rule specification

# ########################################################################
# Rule Specification
# ########################################################################
# Rule Identifier:
#   waf_regional_webacl_not_empty_check
# Description:
#   This control checks whether an AWS WAF Classic Regional web ACL contains any WAF rules
#   or rule groups.
# Reports on:
#   AWS::WAFRegional::WebACL
# Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# Rule Parameters:
#   None
# Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any WAF Classic Regional web ACL resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an AWS WAF Classic Regional web ACL resource
#     And: 'Rules' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an AWS WAF Classic Regional web ACL resource
#     And: 'Rules' has been provided as an empty list
#     Then: FAIL
#   Scenario: 4
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document contains an AWS WAF Classic Regional web ACL resource
#     And: 'Rules' has been provided as a non-empty list
#     Then: PASS
# Constants

let WAF_REGIONAL_WEB_ACL_TYPE = "AWS::WAFRegional::WebACL"
let INPUT_DOCUMENT = this

# Assignments

let waf_regional_web_acls = Resources.[Type == %WAF_REGIONAL_WEB_ACL_TYPE ]

# Primary Rules

rule waf_regional_webacl_not_empty_check when is_cfn_template(%INPUT_DOCUMENT) {  
  %waf_regional_web_acls not empty {  
    check(%waf_regional_web_acls.Properties)  
    <<  
    [CT.WAF-REGIONAL.PR.2]: Require any AWS WAF regional web access control list (ACL) to have a rule or rule group  
    [FIX]: Provide one or more AWS WAF rules within the 'Rules' property.  
    >>  }  
}

rule waf_regional_webacl_not_empty_check when is_cfn_hook(%INPUT_DOCUMENT, %WAF_REGIONAL_WEB_ACL_TYPE) {  
  check(%INPUT_DOCUMENT.%WAF_REGIONAL_WEB_ACL_TYPE.resourceProperties)  
  <<  
  [CT.WAF-REGIONAL.PR.2]: Require any AWS WAF regional web access control list (ACL) to have a rule or rule group  
  [FIX]: Provide one or more AWS WAF rules within the 'Rules' property.  
  >>  }  

# Parameterized Rules

rule check(waf_regional_web_acl) {  
  %waf_regional_web_acl {  
    # Scenario 2, 3 and 4  
    Rules exists  
    Rules is_list  
    Rules not empty  
  }  
}

# Utility Rules

rule is_cfn_template(doc) {  
  %doc {  
    AWSTemplateFormatVersion exists or  
    Resources exists  
  }  
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {  
  %doc.%RESOURCE_TYPE.resourceProperties exists  
}

CT.WAF-REGIONAL.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.
PASS Example - Use this template to verify a compliant resource creation.

Resources:
IPSetDenylist:
  Type: "AWS::WAFRegional::IPSet"
Properties:
  Name: "IPSet for deny listed IP addresses"
  IPSetDescriptors:
  - Type: "IPV4"
    Value: "192.0.2.44/32"
IPSetRule:
  Type: AWS::WAFRegional::Rule
Properties:
  Name: ExampleIPSetRule
  MetricName: ExampleIPSetRuleMetric
  Predicates:
    - DataId:
      Ref: IPSetDenylist
    Negated: false
  Type: IPMatch
WafRegionalWebAcl:
  Type: AWS::WAFRegional::WebACL
Properties:
  Name: ExampleWebACL
  DefaultAction:
    Type: ALLOW
  MetricName: ExampleWebACLMetric
  Rules:
    - Action:
      Type: BLOCK
      Priority: 1
      RuleId:
        Ref: IPSetRule

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
WafRegionalWebAcl:
  Type: AWS::WAFRegional::WebACL
Properties:
  Name: ExampleWebACL
  DefaultAction:
    Type: ALLOW
  MetricName: ExampleWebACLMetric

AWS WAF controls

Topics
- [CT.WAF.PR.1] Require any AWS WAF global rule to have a condition (p. 1324)
- [CT.WAF.PR.2] Require any AWS WAF global web ACL to have a rule or rule group (p. 1328)

[CT.WAF.PR.1] Require any AWS WAF global rule to have a condition

This control checks whether an AWS WAF Classic global rule contains any conditions.
• **Control objective**: Limit network access
• **Implementation**: AWS CloudFormation Guard Rule
• **Control behavior**: Proactive
• **Resource types**: AWS::WAF::Rule
• **AWS CloudFormation guard rule**: [CT.WAF.PR.1 rule specification (p. 1326)](https://aws.amazon.com)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see: [CT.WAF.PR.1 rule specification (p. 1326)](https://aws.amazon.com)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.WAF.PR.1 example templates (p. 1327)](https://aws.amazon.com)

**Explanation**

A WAF global rule can contain multiple conditions. A rule's conditions allow for traffic inspection and take a defined action (allow, block, or count). Without any conditions, the traffic passes without inspection. A WAF global rule with no conditions, but with a name or tag suggesting allow, block, or count, could lead to the wrong assumption that one of those actions is occurring.

**Remediation for rule failure**

Provide one or more AWS WAF rule conditions within the Predicates property.

The examples that follow show how to implement this remediation.

**AWS WAF Classic Global Rule - Example**

AWS WAF Classic global rule configured with an IP match predicate. The example is shown in JSON and YAML.

**JSON example**

```json
{
    "WAFRule": {
        "Type": "AWS::WAF::Rule",
        "Properties": {
            "Name": "SampleWAFRule",
            "MetricName": "SampleWAFRuleMetric",
            "Predicates": [
                {
                    "DataId": {
                        "Ref": "IPSet"
                    },
                    "Negated": false,
                    "Type": "IPMatch"
                }
            ]
        }
    }
}
```

**YAML example**

```yaml
WAFRule:
```
Type: AWS::WAF::Rule
Properties:
  Name: SampleWAFRule
  MetricName: SampleWAFRuleMetric
  Predicates:
    - DataId: !Ref 'IPSet'
      Negated: false
      Type: IPMatch

CT.WAF.PR.1 rule specification

```
# ################################################################
##       Rule Specification       ##
# ################################################################
#
# Rule Identifier:
# waf_global_rule_not_empty_check
#
# Description:
# This control checks whether an AWS WAF Classic global rule contains any conditions.
#
# Reports on:
# AWS::WAF::Rule
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation Hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document does not contain any WAF Classic global rule resources
# Then: SKIP
# Scenario: 2
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a WAF Classic global rule resource
# And: 'Predicates' has not been provided
# Then: FAIL
# Scenario: 3
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a WAF Classic global rule resource
# And: 'Predicates' has been provided as an empty list
# Then: FAIL
# Scenario: 4
# Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# And: The input document contains a WAF Classic global rule resource
# And: 'Predicates' has been provided as a non-empty list
# Then: PASS
#
# Constants
#
let WAF_GLOBAL_RULE_TYPE = "AWS::WAF::Rule"
let INPUTDOCUMENT = this
```
# Assignments

let waf_global_rules = Resources.*[ Type == %WAF_GLOBAL_RULE_TYPE ]

# Primary Rules

rule waf_global_rule_not_empty_check when is_cfn_template(%INPUT_DOCUMENT)
    check(%waf_global_rules.Properties)
    %waf_global_rules not empty {
        [CT.WAF.PR.1]: Require any AWS WAF global rule to have a condition
        [FIX]: Provide one or more AWS WAF rule conditions within the 'Predicates'
        property.
    }

rule waf_global_rule_not_empty_check when is_cfn_hook(%INPUT_DOCUMENT, %WAF_GLOBAL_RULE_TYPE) {
    check(%INPUT_DOCUMENT.%WAF_GLOBAL_RULE_TYPE.resourceProperties)
    %waf_global_rules not empty {
        [CT.WAF.PR.1]: Require any AWS WAF global rule to have a condition
        [FIX]: Provide one or more AWS WAF rule conditions within the 'Predicates'
        property.
    }
}

# Parameterized Rules

rule check(waf_global_rule) {
    %waf_global_rule {
        # Scenario 2, 3 and 4
        Predicates exists
        Predicates is_list
        Predicates not empty
    }
}

# Utility Rules

rule is_cfn_template(doc) {
    %doc {
        AWSTemplateFormatVersion exists or
        Resources exists
    }
}

rule is_cfn_hook(doc, RESOURCE_TYPE) {
    %doc.%RESOURCE_TYPE.resourceProperties exists
}

CT.WAF.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
PROACTIVE CONTROLS

**IPSetDenylist**:
- **Type**: AWS::WAF::IPSet
- **Properties**:
  - **Name**: IPSet for deny listed IP addresses
  - **IPSetDescriptors**:
    - **Type**: IPV4
    - **Value**: 192.0.2.44/32

**WafGlobalRule**:
- **Type**: AWS::WAF::Rule
- **Properties**:
  - **Name**: ExampleWAFRule
  - **MetricName**: ExampleWAFRuleMetric
  - **Predicates**:
    - **DataId**:
      - **Ref**: IPSetDenylist
      - **Negated**: false
      - **Type**: "IPMatch"

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

**Resources**:
- **WafGlobalRule**:
  - **Type**: AWS::WAF::Rule
  - **Properties**:
    - **Name**: ExampleWAFRule
    - **MetricName**: ExampleWAFRuleMetric

**[CT.WAF.PR.2]** Require any AWS WAF global web ACL to have a rule or rule group

This control checks whether an AWS WAF Classic global web ACL contains any WAF rules or rule groups.

- **Control objective**: Limit network access
- **Implementation**: AWS CloudFormation Guard Rule
- **Control behavior**: Proactive
- **Resource types**: AWS::WAF::WebACL
- **AWS CloudFormation guard rule**: [CT.WAF.PR.2 rule specification](p. 1329)

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.WAF.PR.2 rule specification](p. 1329)
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.WAF.PR.2 example templates](p. 1331)

**Explanation**

A WAF global web ACL can contain a collection of rules and rule groups that inspect and control web requests. If a web ACL is empty, the web traffic can pass without being detected or acted upon by WAF depending on the default action.

**Remediation for rule failure**

Provide one or more AWS WAF rules within the Rules property.
The examples that follow show how to implement this remediation.

**AWS WAF Classic Global Web ACL - Example**

AWS WAF Classic global web ACL configured with a rule to block requests based on an IP set match. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "WafGlobalWebAcl": {
        "Type": "AWS::WAF::WebACL",
        "Properties": {
            "Name": "SampleWebACL",
            "DefaultAction": {
                "Type": "ALLOW"
            },
            "MetricName": "SampleWebACLMetric",
            "Rules": [
                {
                    "Action": {
                        "Type": "BLOCK"
                    },
                    "Priority": 1,
                    "RuleId": {
                        "Ref": "IPSetRule"
                    }
                }
            ]
        }
    }
}
```

**YAML example**

```yaml
WafGlobalWebAcl:
  Type: AWS::WAF::WebACL
  Properties:
    Name: SampleWebACL
    DefaultAction:
      Type: ALLOW
    MetricName: SampleWebACLMetric
    Rules:
    - Action:
        Type: BLOCK
        Priority: 1
        RuleId: !Ref 'IPSetRule'
```

**CT.WAF.PR.2 rule specification**

```plaintext
# ****************************
** Rule Specification  **
# ****************************
#
# Rule Identifier:
#  waf_global_webacl_not_empty_check
#
```
# Description:
#   This control checks whether an AWS WAF Classic global web ACL contains any WAF rules or
#   rule groups.
# # Reports on:
#   AWS::WAF::WebACL
# # Evaluates:
#   AWS CloudFormation, AWS CloudFormation hook
# # Rule Parameters:
#   None
# # Scenarios:
#   Scenario: 1
#     Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#     And: The input document does not contain any AWS WAF Classic global web ACL
#         resources
#     Then: SKIP
#   Scenario: 2
#     Given: The input document contains an AWS WAF Classic global web ACL
#         resource
#     And: 'Rules' has not been provided
#     Then: FAIL
#   Scenario: 3
#     Given: The input document contains an AWS WAF Classic global web ACL
#         resource
#     And: 'Rules' has been provided as an empty list
#     Then: FAIL
#   Scenario: 4
#     Given: The input document contains an AWS WAF Classic global web ACL
#         resource
#     And: 'Rules' has been provided as a non-empty list
#     Then: PASS
#
# # Constants
# let WAF_GLOBAL_WEB_ACL_TYPE = "AWS::WAF::WebACL"
# let INPUT_DOCUMENT = this
#
# # Assignments
# let waf_global_web_acls = Resources.*[ Type == WAF_GLOBAL_WEB_ACL_TYPE ]
#
# # Primary Rules
# rule waf_global_webacl_not_empty_check when is_cfn_template(INPUT_DOCUMENT)
#   check(waf_global_web_acls not empty { 
#     [CT.WAF.PR.2]: Require any AWS WAF global web ACL to have a rule or rule group
#     [FIX]: Provide one or more AWS WAF rules within the 'Rules' property.
#   })
#
# rule waf_global_webacl_not_empty_check when is_cfn_hook(INPUT_DOCUMENT,
#   WAF_GLOBAL_WEB_ACL_TYPE) {
#   check(INPUT_DOCUMENT.%WAF_GLOBAL_WEB_ACL_TYPE.resourceProperties)
# }
[CT.WAF.PR.2]: Require any AWS WAF global web ACL to have a rule or rule group

[Fix]: Provide one or more AWS WAF rules within the 'Rules' property.

## Parameterized Rules

```
rule check(waf_global_web_acl) {
  %waf_global_web_acl {
    # Scenario 2, 3 and 4
    Rules exists
    Rules is_list
    Rules not empty
  }
}
```

## Utility Rules

```
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```

### CT.WAF.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
- **IPSetDenylist**:
  - Type: "AWS::WAF::IPSet"
  - Properties:
    - Name: "IPSet for deny listed IP addresses"
    - IPSetDescriptors:
      - Type: "IPV4"
      - Value: "192.0.2.44/32"

- **IPSetRule**:
  - Type: AWS::WAF::Rule
  - Properties:
    - Name: ExampleIPSetRule
    - MetricName: ExampleIPSetRuleMetric
    - Predicates:
      - DataId:
        - Ref: IPSetDenylist
        - Negated: false
        - Type: IPMatch

- **WafGlobalWebAcl**:
  - Type: AWS::WAF::WebACL
  - Properties:
    - Name: ExampleWebACL
    - DefaultAction:
      - Type: ALLOW
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

```
MetricName: ExampleWebACLMetric
Rules:
- Action:
  Type: BLOCK
  Priority: 1
  RuleId:
  Ref: IPSetRule

Resources:
WafGlobalWebAcl:
  Type: AWS::WAF::WebACL
  Properties:
    Name: ExampleWebACL
    DefaultAction:
      Type: ALLOW
      MetricName: ExampleWebACLMetric
```

**AWS WAFV2 controls**

**Topics**
- [CT.WAFV2.PR.1] Require an AWS WAFV2 web ACL to be non-empty (p. 1332)
- [CT.WAFV2.PR.2] Require an AWS WAFV2 rule group to be non-empty (p. 1336)

**[CT.WAFV2.PR.1] Require an AWS WAFV2 web ACL to be non-empty**

This control checks whether an AWS WAFV2 web ACL contains any WAF rules or WAF rule groups.

- **Control objective:** Limit network access
- **Implementation:** AWS CloudFormation Guard Rule
- **Control behavior:** Proactive
- **Resource types:** AWS::WAFv2::WebACL
- **AWS CloudFormation guard rule:** [CT.WAFV2.PR.1 rule specification (p. 1334)]

**Details and examples**

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.WAFV2.PR.1 rule specification (p. 1334)]
- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.WAFV2.PR.1 example templates (p. 1336)]

**Explanation**

A web access control list (ACL) attached to WAFv2 can contain a collection of rules and rule groups. The rules are designed to inspect and control web requests. If a web ACL is empty, the web traffic passes without being detected or acted upon by the web application firewall (WAF).

**Remediation for rule failure**

Provide one or more AWS WAFV2 rules within the Rules property.
The examples that follow show how to implement this remediation.

**AWS WAFV2 web ACL - Example**

AWS WAFV2 web ACL configured with a rule to block requests based on a cross-site scripting (XSS) match. statement. The example is shown in JSON and in YAML.

**JSON example**

```json
{
    "WAFv2WebACL": {
        "Type": "AWS::WAFv2::WebACL",
        "Properties": {
            "Scope": "REGIONAL",
            "Description": "Sample WebACL",
            "DefaultAction": {
                  "Allow": {}
            },
            "VisibilityConfig": {
                "SampledRequestsEnabled": true,
                "CloudWatchMetricsEnabled": true,
                "MetricName": "SampleWebACLMetric"
            },
            "Rules": [
                {
                    "Name": "SampleXssRule",
                    "Priority": 0,
                    "Action": {
                         "Block": {}
                    },
                    "VisibilityConfig": {
                        "SampledRequestsEnabled": true,
                        "CloudWatchMetricsEnabled": true,
                        "MetricName": "SampleXssMatchMetric"
                    },
                    "Statement": {
                        "XssMatchStatement": {
                            "FieldToMatch": {
                                "AllQueryArguments": {}
                            },
                            "TextTransformations": [
                                {
                                    "Priority": 1,
                                    "Type": "NONE"
                                }
                            ]
                        }
                    }
                }
            ]
        }
    }
}

**YAML example**

```
WAFv2WebACL:
  Type: AWS::WAFv2::WebACL
  Properties:
    Scope: REGIONAL
```
Description: Sample WebACL
DefaultAction:
  Allow: {}
VisibilityConfig:
  SampledRequestsEnabled: true
  CloudWatchMetricsEnabled: true
  MetricName: SampleWebACLMetric
Rules:
  - Name: SampleXssRule
    Priority: 0
    Action:
      Block: {}
    VisibilityConfig:
      SampledRequestsEnabled: true
      CloudWatchMetricsEnabled: true
      MetricName: SampleXssMatchMetric
Statement:
  XssMatchStatement:
    FieldToMatch:
      AllQueryArguments: {}
    TextTransformations:
      - Priority: 1
        Type: NONE

CT.WAFV2.PR.1 rule specification

# #####################################################################
## Rule Specification
# #####################################################################
# # Rule Identifier:
# # wafv2_webacl_not_empty_check
# # Description:
# # This control checks whether an AWS WAFV2 web ACL contains any WAF rules or WAF rule groups.
# # Reports on:
# # AWS::WAFv2::WebACL
# # Evaluates:
# # AWS CloudFormation, AWS CloudFormation hook
# # Rule Parameters:
# # None
# # Scenarios:
# # Scenario: 1
# # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# # And: The input document does not contain any WAFv2 web ACL resources
# # Then: SKIP
# # Scenario: 2
# # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# # And: The input document contains an WAFv2 web ACL resource
# # And: 'Rules' has not been provided
# # Then: FAIL
# # Scenario: 3
# # Given: The input document is an AWS CloudFormation or AWS CloudFormation hook document
# Constants

```plaintext
let WAFV2_WEB_ACL_TYPE = "AWS::WAFv2::WebACL"
let INPUT_DOCUMENT = this
```

# Assignments

```plaintext
let wafv2_web_acls = Resources.*[ Type == %WAFV2_WEB_ACL_TYPE ]
```

# Primary Rules

```plaintext
rule wafv2_web_acl_not_empty_check when is_cfn_template(%INPUT_DOCUMENT) {
  %wafv2_web_acls not empty {
    check(%wafv2_web_acls.Properties)
    [CT.WAFV2.PR.1]: Require an AWS WAFV2 web ACL to be non-empty
    [FIX]: Provide one or more AWS WAFv2 rules within the 'Rules' property.
  }
}
rule wafv2_web_acl_not_empty_check when is_cfn_hook(%INPUT_DOCUMENT, %WAFV2_WEB_ACL_TYPE) {
  check(%INPUT_DOCUMENT.%WAFV2_WEB_ACL_TYPE.resourceProperties)
  [CT.WAFV2.PR.1]: Require an AWS WAFV2 web ACL to be non-empty
  [FIX]: Provide one or more AWS WAFv2 rules within the 'Rules' property.
}
```

# Parameterized Rules

```plaintext
rule check(wafv2_web_acl) {
  %wafv2_web_acl {
    # Scenario 2, 3 and 4
    Rules exists
    Rules is_list
    Rules not empty
  }
}
```

# Utility Rules

```plaintext
rule is_cfn_template(doc) {
  %doc {
    AWSTemplateFormatVersion exists or
    Resources exists
  }
}
rule is_cfn_hook(doc, RESOURCE_TYPE) {
  %doc.%RESOURCE_TYPE.resourceProperties exists
}
```
CT.WAFV2.PR.1 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

PASS Example - Use this template to verify a compliant resource creation.

Resources:
WAFv2WebACL:
  Type: AWS::WAFv2::WebACL
  Properties:
    Scope: REGIONAL
    Description: Example WebACL
    DefaultAction:
      Allow: {}
    VisibilityConfig:
      SampledRequestsEnabled: true
      CloudWatchMetricsEnabled: true
      MetricName: ExampleWebACLMetric
    Rules:
      - Name: ExampleXssRule
        Priority: 0
        Action:
          Block: {}
        VisibilityConfig:
          SampledRequestsEnabled: true
          CloudWatchMetricsEnabled: true
          MetricName: ExampleXssMatchMetric
        Statement:
          XssMatchStatement:
            FieldToMatch:
              AllQueryArguments: {}
            TextTransformations:
              - Priority: 1
                Type: NONE

FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
WAFv2WebACL:
  Type: AWS::WAFv2::WebACL
  Properties:
    Scope: REGIONAL
    Description: Example WebACL
    DefaultAction:
      Allow: {}
    VisibilityConfig:
      SampledRequestsEnabled: true
      CloudWatchMetricsEnabled: true
      MetricName: ExampleWebACLMetric

[CT.WAFV2.PR.2] Require an AWS WAFV2 rule group to be non-empty

This control checks whether AWS WAFV2 rule groups contain rules.
• **Control objective:** Limit network access
  
  **Implementation:** AWS CloudFormation Guard Rule
  
  **Control behavior:** Proactive
  
  **Resource types:** AWS::WAFv2::RuleGroup
  
  **AWS CloudFormation guard rule:** [CT.WAFV2.PR.2 rule specification (p. 1338)](https://docs.aws.amazon.com/wafv2/latest/developerguide/ctwafv2pr2-rule-specification.html)

## Details and examples

- For details about the PASS, FAIL, and SKIP behaviors associated with this control, see the: [CT.WAFV2.PR.2 rule specification (p. 1338)](https://docs.aws.amazon.com/wafv2/latest/developerguide/ctwafv2pr2-rule-specification.html)

- For examples of PASS and FAIL CloudFormation Templates related to this control, see: [CT.WAFV2.PR.2 example templates (p. 1340)](https://docs.aws.amazon.com/wafv2/latest/developerguide/ctwafv2pr2-example-templates.html)

## Explanation

An AWS WAFV2 rule group can contain multiple rules. The rules are designed to inspect and control web requests. When an empty AWS WAFV2 rule group is associated with a web ACL, and the web ACL is not associated with any other rules or rule groups, the web traffic passes without being detected or acted upon by the web application firewall (WAF).

### Remediation for rule failure

Provide one or more AWS WAFV2 rules within the `Rules` property.

The examples that follow show how to implement this remediation.

#### AWS WAFV2 Rule Group - Example

AWS WAFV2 rule group configured with a rule to block requests based on a cross-site scripting (XSS) match statement. The example is shown in JSON and in YAML.

#### JSON example

```json
{
  "WAFv2RuleGroup": {
    "Type": "AWS::WAFv2::RuleGroup",
    "Properties": {
      "Scope": "REGIONAL",
      "Description": "Sample Rule Group",
      "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "SampleRuleGroupMetric"
      },
      "Capacity": 1000,
      "Rules": [
        {
          "Name": "XssRule",
          "Priority": 0,
          "Action": {
            "Block": {}
          },
          "VisibilityConfig": {
            "SampledRequestsEnabled": true,
            "CloudWatchMetricsEnabled": true,
            "MetricName": "SampleXssMatchMetric"
          },
          "Statement": {
            "XssMatchStatement": {
```
YAML example

```yaml
WAFv2RuleGroup:
  Type: AWS::WAFv2::RuleGroup
  Properties:
    Scope: REGIONAL
    Description: Sample Rule Group
    VisibilityConfig:
      SampledRequestsEnabled: true
      CloudWatchMetricsEnabled: true
      MetricName: SampleRuleGroupMetric
    Capacity: 1000
  Rules:
    - Name: XssRule
      Priority: 0
      Action:
        Block: {}
      VisibilityConfig:
        SampledRequestsEnabled: true
        CloudWatchMetricsEnabled: true
        MetricName: SampleXssMatchMetric
    Statement:
      XssMatchStatement:
        FieldToMatch:
          AllQueryArguments: {}
        TextTransformations:
          - Priority: 1
            Type: NONE
```

CT.WAFV2.PR.2 rule specification

```plaintext
# ###################################################################
# Rule Specification  ##
# ###################################################################
#
# Rule Identifier:
# wafv2_rulegroup_not_empty_check
#
# Description:
# This control checks whether AWS WAFV2 rule groups contain rules.
#```
# Reports on:
# AWS::WAFv2::RuleGroup
#
# Evaluates:
# AWS CloudFormation, AWS CloudFormation hook
#
# Rule Parameters:
# None
#
# Scenarios:
# Scenario: 1
#   Given: The input document is an AWS CloudFormation or AWS CloudFormation hook
document
#         And: The input document does not contain any AWS WAFV2 rule group resources
#         Then: SKIP
# Scenario: 2
#   Given: The input document contains an AWS WAFV2 rule group resource
#         And: 'Rules' has not been provided
#         Then: FAIL
# Scenario: 3
#   Given: The input document contains an AWS WAFV2 rule group resource
#         And: 'Rules' has been provided as an empty list
#         Then: FAIL
# Scenario: 4
#   Given: The input document contains an AWS WAFV2 rule group resource
#         And: 'Rules' has been provided as a non-empty list
#         Then: PASS
#
# Constants
#
let WAFV2_RULE_GROUP_TYPE = "AWS::WAFv2::RuleGroup"
let INPUT_DOCUMENT = this
#
# Assignments
#
let wafv2_rule_groups = Resources.*[ Type == %WAFV2_RULE_GROUP_TYPE ]
#
# Primary Rules
#
rule wafv2_rulegroup_not_empty_check when is_cfn_template(%INPUT_DOCUMENT)
%wafv2_rule_groups not empty {
  check(%wafv2_rule_groups.Properties)
  <<
    [CT.WAFV2.PR.2]: Require an AWS WAFV2 rule group to be non-empty
    [FIX]: Provide one or more AWS WAFV2 rules within the 'Rules' property.
  >>
}

rule wafv2_rulegroup_not_empty_check when is_cfn_hook(%INPUT_DOCUMENT,
%WAFV2_RULE_GROUP_TYPE) {
  check(%INPUT_DOCUMENT.%WAFV2_RULE_GROUP_TYPE.resourceProperties)
  <<
    [CT.WAFV2.PR.2]: Require an AWS WAFV2 rule group to be non-empty
    [FIX]: Provide one or more AWS WAFV2 rules within the 'Rules' property.
  >>
}
# Parameterized Rules

```haskell
# Scenario 2, 3 and 4
%wafv2_rule_group {
    Rules exists
    Rules is_list
    Rules not empty
}
```

# Utility Rules

```haskell
# Scenario 2, 3 and 4
%wafv2_rule_group {
    Rules exists
    Rules is_list
    Rules not empty
}
```

## CT.WAFV2.PR.2 example templates

You can view examples of the PASS and FAIL test artifacts for the AWS Control Tower proactive controls.

**PASS Example** - Use this template to verify a compliant resource creation.

```json
Resources:
WAFv2RuleGroup:
  Type: AWS::WAFv2::RuleGroup
  Properties:
    Scope: REGIONAL
    Description: Example Rule Group
    VisibilityConfig:
      SampledRequestsEnabled: true
      CloudWatchMetricsEnabled: true
      MetricName: SampleRuleGroupMetrics
    Capacity: 1000
    Rules:
    - Name: TestXssRule
      Priority: 0
      Action:
        Block: {}
      VisibilityConfig:
        SampledRequestsEnabled: true
        CloudWatchMetricsEnabled: true
        MetricName: ExampleXssMatchMetric
    Statement:
      XssMatchStatement:
        FieldToMatch:
          AllQueryArguments: {}
        TextTransformations:
          - Priority: 1
            Type: NONE
```

```
FAIL Example - Use this template to verify that the control prevents non-compliant resource creation.

Resources:
WAFv2RuleGroup:
  Type: AWS::WAFv2::RuleGroup
  Properties:
  Description: Example Rule Group
  VisibilityConfig:
    SampledRequestsEnabled: true
    CloudWatchMetricsEnabled: true
    MetricName: ExampleRuleGroupMetric
  Capacity: 1000
  Rules: []

Security Hub standard

AWS Control Tower is integrated with AWS Security Hub to provide detective controls that help you monitor your AWS environment. The integration is accomplished with a Security Hub standard, called the Service-Managed Standard: AWS Control Tower.


This standard is available only for AWS Control Tower customers who have created the standard in the AWS Control Tower console. AWS Control Tower creates the standard for you when you enable the first Security Hub control in the AWS Control Tower console. When you enable the first control, if you haven’t already enabled Security Hub, AWS Control Tower also enables Security Hub for you.

After you create this standard, you can view the Security Hub detective controls alongside other AWS Control Tower controls, in the AWS Control Tower console and in Security Hub.

Control behavior

- No controls are enabled automatically when you create this standard in AWS Control Tower.
- The Security Hub controls are active at the OU level only, not for all AWS Control Tower OUs (if not enabled for all), and not for individual accounts.
- When Security Hub adds new controls, the new controls are not added to the Service-Managed Standard: AWS Control Tower automatically.

Enable or remove controls for the Service-Managed Standard

To avoid drift, always enable and remove controls for the Service-Managed Standard by means of the AWS Control Tower service, either in the console or by calling the AWS Control Tower APIs, EnableControl and DisableControl. When you change the enablement status of a control in AWS Control Tower, the change also is reflected in Security Hub.

Note
If you deactivate a Service-Managed Standard control by means of the Security Hub console, the AWS Control Tower member account enters a state of control drift. In this situation, AWS
Control Tower is not receiving the Security Hub findings for the control that you deactivated. You must resolve this drift before AWS Control Tower can apply the control successfully to your registered organizational units and member accounts.

You can delete this standard in the AWS Control Tower console by deactivating all controls in the standard. This deletes the standard for all managed accounts and governed Regions in AWS Control Tower. Deleting the standard does not deactivate Security Hub for your account.

**Deprecated control**
The control named [SH.S3.4] S3 buckets should have server-side encryption enabled is deprecated, effective July 18, 2023. It was removed from the controls library on August 18, 2023. For more information, see [AWS Control Tower deprecates two controls](p. 1462).

**Security Hub score and findings**
Based on control status, Security Hub calculates a security score for the **Service-Managed Standard: AWS Control Tower**. This security score and the control findings are available only in Security Hub. These items aren't available in AWS Control Tower.

**Note**
When you create **Service-Managed Standard: AWS Control Tower** and enable controls for it, Security Hub may take up to 18 hours to generate findings for controls that use the same underlying AWS Config service-linked rule as controls from other enabled Security Hub standards. For more information, see [Schedule for running security checks](p. 1342) in the AWS Security Hub User Guide.

**Security Hub control drift reporting**
When reporting drift for controls that are part of the AWS Security Hub Service-Managed Standard, AWS Control Tower receives a daily status update from Security Hub. If no update is received, AWS Control Tower verifies whether drift has occurred. If so, AWS Control Tower reports drift. If a control shows drift, AWS Control Tower sends an Amazon SNS notification to the AWS Control Tower security-aggregate-notification channel. You must subscribe to this SNS notification to receive information about which specific account is affected by Security Hub control drift. For more information about Security Hub control drift in AWS Control Tower, see [Security Hub control drift](p. 178).

Although controls are active in every governed Region, AWS Control Tower sends the AWS Security Hub **Finding** events to the AWS Control Tower home Region only.

**Remediate drift**
When drift is reported, you can remediate the situation by choosing **Re-register OU** on the **Organizations** page in the AWS Control Tower console, or by deactivating and re-activating the control that's in a drifted state, either by means of the console, or through the AWS Control Tower API.

**Note**
You can enable and manage some Security Hub controls from AWS Control Tower, with the **Security Hub Service-managed Standard: AWS Control Tower**. It is important to know that some Security Hub controls do not operate in certain AWS Regions where AWS Control Tower is available, because those Regions do not support the required underlying functionality. As a result, when you deploy an Security Hub control through AWS Control Tower, the control may not be operating in all Regions that you govern with AWS Control Tower. For more information about the Security Hub controls that cannot be deployed in certain Regions, see the [Security Hub controls reference documentation](p. 1342).

**The following AWS Regions do not support controls that are part of the Security Hub Service-managed Standard: AWS Control Tower**

- Asia Pacific (Hong Kong) Region, ap-east-1
Data residency controls

- Asia Pacific (Jakarta) Region, ap-southeast-3
- Asia Pacific (Osaka) Region, ap-northeast-3
- Europe (Milan) Region, eu-south-1
- Africa (Cape Town) Region, af-south-1
- Middle East (Bahrain) Region, me-south-1
- Israel (Tel Aviv), il-central-1
- Middle East (UAE) Region, me-central-1
- Europe (Spain) Region, eu-south-2
- Asia Pacific (Hyderabad) Region, ap-south-2
- Europe (Zurich) Region, eu-central-2
- Asia Pacific (Melbourne) Region ap-southeast-4

You can view the Regions for each control in the AWS Control Tower console.

Controls that enhance data residency protection

These elective controls complement your enterprise’s data residency posture. By applying these controls together, you can set up your multi-account environment to help detect and inhibit the purposeful or accidental creation, sharing, or copying of data, outside of your selected AWS Region or Regions.

These controls take effect at the OU level, and they apply to all member accounts within the OU.

Important
Certain global AWS services, such as AWS Identity and Access Management (IAM) and AWS Organizations, are exempt from these controls. You can identify the services that are exempt by reviewing the Region deny SCP, shown in the example code. Services with "*" after their identifier are exempt, because all actions are permitted when the "*" notation is given. This SCP essentially contains a list of explicitly permitted actions, and all other actions are denied. You cannot deny access to your home Region.

Video: Enable data residency controls

This video (5:58) describes how to enable data residency controls with AWS Control Tower controls. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Enabling Data Residency Controls in AWS Control Tower.

Note
We are transitioning our terminology to align better with industry usage and with other AWS services. During this time, you may see the previous term, guardrail, as well as the new term, control, in our documentation, console, blogs, and videos. These terms are synonymous for our purposes.

Topics
- Deny access to AWS based on the requested AWS Region (p. 1344)
- Disallow internet access for an Amazon VPC instance managed by a customer (p. 1346)
- Disallow Amazon Virtual Private Network (VPN) connections (p. 1347)
- Disallow cross-region networking for Amazon EC2, Amazon CloudFront, and AWS Global Accelerator (p. 1347)
- Detect whether public IP addresses for Amazon EC2 autoscaling are enabled through launch configurations (p. 1348)
• Detect whether replication instances for AWS Database Migration Service are public (p. 1349)
• Detect whether Amazon EBS snapshots are restorable by all AWS accounts (p. 1350)
• Detect whether any Amazon EC2 instance has an associated public IPv4 address (p. 1351)
• Detect whether Amazon S3 settings to block public access are set as true for the account (p. 1351)
• Detects whether an Amazon EKS endpoint is blocked from public access (p. 1352)
• Detect whether an Amazon OpenSearch Service domain is in Amazon VPC (p. 1353)
• Detect whether any Amazon EMR cluster master nodes have public IP addresses (p. 1354)
• Detect whether the AWS Lambda function policy attached to the Lambda resource blocks public access (p. 1355)
• Detect whether public routes exist in the route table for an Internet Gateway (IGW) (p. 1356)
• Detect whether Amazon Redshift clusters are blocked from public access (p. 1357)
• Detect whether an Amazon SageMaker notebook instance allows direct internet access (p. 1357)
• Detect whether any Amazon VPC subnets are assigned a public IP address (p. 1358)
• Detect whether AWS Systems Manager documents owned by the account are public (p. 1359)

Deny access to AWS based on the requested AWS Region

This control is commonly referred to as the Region deny control.

This control disallows access to unlisted operations in global and regional services outside of the specified Regions. That includes all Regions where AWS Control Tower is not available, as well as all Regions not selected for governance in the Landing zone settings page. Actions are allowed as usual in Regions with Governed status.

Note
Certain global AWS services, such as AWS Identity and Access Management (IAM) and AWS Organizations, are exempt from data residency controls. Those services are specified in the SCP example code that follows.

This is an elective control with preventive guidance. It is the primary control associated with the Region deny action. For more information, see Configure the Region deny control (p. 102).

The format for this control is based on the following SCP.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRREGIONDENY",
         "Effect": "Deny",
         "NotAction": [
            "a4b:*",
            "access-analyzer:*",
            "account:*",
            "acm:*",
            "activate:*",
            "artifact:*",
            "aws-marketplace-management:*",
            "aws-marketplace:*",
            "aws-portal:*",
            "billing:*",
            "billingconductor:*",
            "budgets:*",
            "ce:*",
            "chatbot:*",
            "chime:*",
```
Data residency controls

"cloudfront:**",
"cloudtrail:LookupEvents",
"compute-optimizer:**",
"config:**",
"consoleapp:**",
"consolidatedbilling:**",
"cur:**",
"datapipeline:GetAccountLimits",
"devicefarm:**",
"directconnect:**",
"discovery-marketplace:**",
"ec2:DescribeRegions",
"ec2:DescribeTransitGateways",
"ec2:DescribeVpnGateways",
"ecr-public:**",
"fms:**",
"freetier:**",
"globalaccelerator:**",
"health:**",
"iam:**",
"importexport:**",
"invoicing:**",
"iq:**",
"kms:**",
"license-manager:ListReceivedLicenses",
"lightsail:Get**",
"mobileanalytics:**",
"networkmanager:**",
"notifications-contacts:**",
"notifications:**",
"organizations:**",
"payments:**",
"pricing:**",
"resource-explorer-2:**",
"route53-recovery-cluster:**",
"route53-recovery-control-config:**",
"route53-recovery-readiness:**",
"route53:**",
"route53domains:**",
"s3:CreateMultiRegionAccessPoint",
"s3:DeleteMultiRegionAccessPoint",
"s3:DescribeMultiRegionAccessPointOperation",
"s3:GetAccountPublicAccessBlock",
"s3:GetBucketLocation",
"s3:GetBucketPolicyStatus",
"s3:GetBucketPublicAccessBlock",
"s3:GetMultiRegionAccessPoint",
"s3:GetMultiRegionAccessPointPolicy",
"s3:GetMultiRegionAccessPointPolicyStatus",
"s3:GetStorageLensConfiguration",
"s3:GetStorageLensDashboard",
"s3:ListAllMyBuckets",
"s3:ListMultiRegionAccessPoints",
"s3:ListStorageLensConfigurations",
"s3:PutAccountPublicAccessBlock",
"s3:PutMultiRegionAccessPointPolicy",
"savingsplans:**",
"shield:**",
"sso:**",
"sts:**",
"support:**",
"supportapp:**",
"supportplans:**",
"sustainability:**",
"tag:GetResources",
"tax:****,
Based on this example SCP format, AWS Control Tower adds your governed Regions into the `aws:RequestedRegion` statement. You cannot exclude your home Region. Actions not listed in the SCP are not permitted.

**Disallow internet access for an Amazon VPC instance managed by a customer**

This control disallows internet access for an Amazon Virtual Private Cloud (VPC) instance managed by a customer, rather than by an AWS service.

*Important*

If you provision Account Factory accounts with VPC internet access settings enabled, that Account Factory setting overrides this control. To avoid enabling internet access for newly provisioned accounts, you must change the setting in Account Factory. For more information, see [Walkthrough: Configure AWS Control Tower Without a VPC](p. 1432).

- This control does not apply to VPCs managed by AWS services.
- Existing VPCs that have internet access retain their internet access. It applies to new instances only. After this control is applied, access cannot be changed.

This is a preventive control with elective guidance. By default, this control isn't enabled on any OUs.

The artifact for this control is the following service control policy (SCP).

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRDISALLOWVPCINTERNETACCESS",
      "Effect": "Deny",
      "Action": [
        "ec2:CreateInternetGateway",
        "ec2:AttachInternetGateway",
        "ec2:CreateEgressOnlyInternetGateway",
        "ec2:AttachEgressOnlyInternetGateway",
        "ec2:CreateDefaultVpc",
        "ec2:CreateDefaultSubnet",
        "ec2:CreateCarrierGateway"
      ],
    }
  ]
}
```
Disallow Amazon Virtual Private Network (VPN) connections

This control prevents Virtual Private Network (VPN) connections (Site-to-Site VPN and Client VPN) to an Amazon Virtual Private Cloud (VPC).

Note
Existing VPCs that have internet access retain their internet access.

This is a preventive control with elective guidance. By default, this control isn't enabled on any OUs.

The artifact for this control is the following service control policy (SCP).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRDISALLOWVPNCONNECTIONS",
            "Effect": "Deny",
            "Action": [
                "ec2:CreateVPNGateway",
                "ec2:AttachVPNGateway",
                "ec2:CreateCustomerGateway",
                "ec2:CreateVpnConnection",
                "ec2:ModifyVpnConnection",
                "ec2:CreateClientVpnEndpoint",
                "ec2:ModifyClientVpnEndpoint",
                "ec2:AssociateClientVpnTargetNetwork",
                "ec2:AuthorizeClientVpnIngress"
            ],
            "Resource": [
                "*
            ]
        }
    ]
}
```

Disallow cross-region networking for Amazon EC2, Amazon CloudFront, and AWS Global Accelerator

This control prevents configuring cross-region networking connections from Amazon EC2, Amazon CloudFront, and AWS Global Accelerator services. It prevents VPC peering and transit gateway peering.

Note
This control prevents Amazon EC2 VPC peering and Amazon EC2 transit gateway peering within a single Region, as well as across Regions. For this reason, this control may affect certain workloads in addition to your data residency posture.
This is a preventive control with elective guidance. By default, this control isn't enabled on any OUs.

The artifact for this control is the following service control policy (SCP).

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRDISALLOWCROSSREGIONNETWORKING",
         "Effect": "Deny",
         "Action": ["ec2:CreateVpcPeeringConnection",
                    "ec2:AcceptVpcPeeringConnection",
                    "ec2:CreateTransitGatewayPeeringAttachment",
                    "ec2:AcceptTransitGatewayPeeringAttachment",
                    "cloudfront:CreateDistribution",
                    "cloudfront:UpdateDistribution",
                    "globalaccelerator:Create*",
                    "globalaccelerator:Update*"
                  ],
         "Resource": ["*"],
      }
   ]
}
```

### Detect whether public IP addresses for Amazon EC2 autoscaling are enabled through launch configurations

This control detects whether Amazon EC2 Auto Scaling groups have public IP addresses enabled through launch configurations.

This is a detective control with elective guidance. By default, this control isn't enabled on any OUs.

**In the console:**

- The rule shows **Non-compliant** status if the launch configuration for an autoscaling group sets the value of the field `AssociatePublicIpAddress` set as **True**.

The artifact for this control is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rule to detect whether public IP addresses for Amazon EC2 Auto Scaling are enabled through launch configurations

Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'

Resources:
  AutoscalingLaunchConfigPublicIpDisabled:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Detects whether Amazon EC2 Auto Scaling groups have public IP addresses enabled through launch configurations. This rule is NON_COMPLIANT if the launch configuration for an Auto Scaling group has the value of the field `AssociatePublicIpAddress` set as True.
```
Scope:

ComplianceResourceTypes:
- AWS::AutoScaling::LaunchConfiguration

Source:

Owner: AWS
SourceIdentifier: AUTOSCALING_LAUNCH_CONFIG_PUBLIC_IP_DISABLED

Detect whether replication instances for AWS Database Migration Service are public

This control detects whether AWS Database Migration Service replication instances are public.

This is a detective control with elective guidance. By default, this control isn't enabled on any OUs.

In the console:

- The rule shows **Non-compliant** status if the value of the `PubliclyAccessible` field is set as **True**.

The artifact for this control is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rule to detect whether replication instances for AWS Database Migration Service are public

Parameters:
ConfigRuleName:
  Type: 'String'
  Description: 'Name for the Config rule'

MaximumExecutionFrequency:
  Type: String
  Default: 24hours
  Description: The frequency at which AWS Config will run evaluations for the rule.
  AllowedValues:
    - 1hour
    - 3hours
    - 6hours
    - 12hours
    - 24hours

Mappings:
  Settings:
    FrequencyMap:
      1hour : One_Hour
      3hours : Three_Hours
      6hours : Six_Hours
      12hours : Twelve_Hours
      24hours : TwentyFour_Hours

Resources:
DmsReplicationNotPublic:
  Type: AWS::Config::ConfigRule
  Properties:
    ConfigRuleName: !Sub ${ConfigRuleName}
    Description: Detects whether AWS Database Migration Service replication instances are public. The rule is NON_COMPLIANT if the value of the `PubliclyAccessible` field is set as True.
    Source:
      Owner: AWS
      SourceIdentifier: DMS_REPLICATION_NOT_PUBLIC
      MaximumExecutionFrequency:
```
Detect whether Amazon EBS snapshots are restorable by all AWS accounts

This control detects whether all AWS accounts have access to restore Amazon EBS snapshots.

This is a detective control with elective guidance. By default, this control isn’t enabled on any OUs.

In the console:

- The rule shows **Non-compliant** status if any snapshots have the RestorableByUserIds field set to the value **All**. In that case, the Amazon EBS snapshots are public.

The artifact for this control is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rule to detect whether Amazon EBS snapshots are restorable by all AWS accounts

Parameters:
ConfigRuleName:
  Type: 'String'
  Description: 'Name for the Config rule'

MaximumExecutionFrequency:
  Type: String
  Default: 24hours
  Description: The frequency at which AWS Config will run evaluations for the rule.
  AllowedValues:
    - 1hour
    - 3hours
    - 6hours
    - 12hours
    - 24hours

Mappings:
Settings:
  FrequencyMap:
    1hour   : One_Hour
    3hours  : Three_Hours
    6hours  : Six_Hours
    12hours : Twelve_Hours
    24hours : TwentyFour_Hours

Resources:
EbsSnapshotPublicRestorableCheck:
  Type: AWS::Config::ConfigRule
  Properties:
    ConfigRuleName: !Sub ${ConfigRuleName}
    Description: Detects whether all AWS accounts have access to restore Amazon EBS snapshots. The rule is NON_COMPLIANT if any snapshots have the RestorableByUserIds field set to the value All. In that case, the Amazon EBS snapshots are public.
    Source:
      Owner: AWS
      SourceIdentifier: EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK
    MaximumExecutionFrequency:
```
Detect whether any Amazon EC2 instance has an associated public IPv4 address

This control detects whether an Amazon Elastic Compute Cloud (Amazon EC2) instance has an associated public IPv4 address. This control applies only to IPv4 addresses.

This is a detective control with elective guidance. By default, this control isn't enabled on any OUs.

In the console:

- The rule shows **Non-compliant** status if the public IP field is present in the Amazon EC2 instance configuration item.

The artifact for this control is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rule to detect whether any Amazon EC2 instance has an associated public IPv4 address
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  Ec2InstanceNoPublicIp:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Detects whether an Amazon Elastic Compute Cloud (Amazon EC2) instance has an associated public IPv4 address. The rule is NON_COMPLIANT if the public IP field is present in the Amazon EC2 instance configuration item.
      Scope:
        ComplianceResourceTypes:
        - AWS::EC2::Instance
      Source:
        Owner: AWS
        SourceIdentifier: EC2_INSTANCE_NO_PUBLIC_IP
```

Detect whether Amazon S3 settings to block public access are set as true for the account

This control periodically detects whether the required Amazon S3 settings to block public access are configured as true for the account, rather than for a bucket or an access point.

In the console:

- The rule shows **Non-compliant** status if at least one of the settings is false.

This is a detective control with elective guidance. By default, this control isn't enabled on any OUs.

The artifact for this control is the following AWS Config rule.
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rule to check whether Amazon S3 settings to block public access are set as true for the account.

Parameters:
ConfigRuleName:
  Type: 'String'
  Description: 'Name for the Config rule'
PublicAccessBlockSetting:
  Type: 'String'
  Default: 'True'
MaximumExecutionFrequency:
  Type: String
  Default: 24hours
  Description: The frequency at which AWS Config will run evaluations for the rule.
  AllowedValues:
    - 1hour
    - 3hours
    - 6hours
    - 12hours
    - 24hours

Mappings:
Settings:
  FrequencyMap:
    1hour   : One_Hour
    3hours  : Three_Hours
    6hours  : Six_Hours
    12hours : Twelve_Hours
    24hours : TwentyFour_Hours

Resources:
CheckForS3PublicAccessBlock:
  Type: AWS::Config::ConfigRule
  Properties:
    ConfigRuleName: !Sub ${ConfigRuleName}
  Description: Checks the Amazon S3 settings to block public access are set as true for the account. The rule is non-compliant if at-least one of the settings is false.
  Source:
    Owner: AWS
    SourceIdentifier: S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC
  Scope:
    ComplianceResourceTypes:
      - AWS::S3::AccountPublicAccessBlock
  InputParameters:
    IgnorePublicAcls: !Ref PublicAccessBlockSetting
    BlockPublicPolicy: !Ref PublicAccessBlockSetting
    BlockPublicAcls: !Ref PublicAccessBlockSetting
    RestrictPublicBuckets: !Ref PublicAccessBlockSetting
  MaximumExecutionFrequency:
    !FindInMap
      - Settings
      - FrequencyMap
      - !Ref MaximumExecutionFrequency

Detects whether an Amazon EKS endpoint is blocked from public access

This control detects whether an Amazon Elastic Kubernetes Service (Amazon EKS) endpoint is blocked from public access.

This is a detective control with elective guidance. By default, this control isn’t enabled on any OUs.
In the console:

- The rule shows **Non-compliant** status if the endpoint is publicly accessible.

The artifact for this control is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rule to detect whether an Amazon EKS endpoint is blocked from public access

Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'

  MaximumExecutionFrequency:
    Type: String
    Default: 24hours
    Description: The frequency at which AWS Config will run evaluations for the rule.
  AllowedValues:
    - 1hour
    - 3hours
    - 6hours
    - 12hours
    - 24hours

Mappings:
  Settings:
    FrequencyMap:
      1hour   : One_Hour
      3hours  : Three_Hours
      6hours  : Six_Hours
      12hours : Twelve_Hours
      24hours : TwentyFour_Hours

Resources:
EKSEndpointNoPublicAccess:
  Type: AWS::Config::ConfigRule
  Properties:
    ConfigRuleName: !Sub ${ConfigRuleName}
    Description: Detects whether an Amazon Elastic Kubernetes Service (Amazon EKS) endpoint is publicly accessible. The rule is NON_COMPLIANT if the endpoint is publicly accessible.
    Source:
      Owner: AWS
      SourceIdentifier: EKS_ENDPOINT_NO_PUBLIC_ACCESS
      MaximumExecutionFrequency:
        !FindInMap
        - Settings
        - FrequencyMap
        - !Ref MaximumExecutionFrequency
```

**Detect whether an Amazon OpenSearch Service domain is in Amazon VPC**

This control detects whether an Amazon OpenSearch Service domain is in Amazon VPC.

This is a detective control with elective guidance. By default, this control isn't enabled on any OUs.

In the console:

- The rule shows **Non-compliant** status if the OpenSearch Service domain endpoint is public.
The artifact for this control is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rule to detect whether an Amazon OpenSearch Service domain is in Amazon VPC

Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'

  MaximumExecutionFrequency:
    Type: String
    Default: 24hours
    Description: The frequency at which AWS Config will run evaluations for the rule.
    AllowedValues:
      - 1hour
      - 3hours
      - 6hours
      - 12hours
      - 24hours

Mappings:
  Settings:
    FrequencyMap:
      1hour : One_Hour
      3hours : Three_Hours
      6hours : Six_Hours
      12hours : Twelve_Hours
      24hours : TwentyFour_Hours

Resources:
  ElasticsearchInVpcOnly:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Detects whether Amazon OpenSearch Service domains are in Amazon Virtual Private Cloud (Amazon VPC). The rule is NON_COMPLIANT if the OpenSearch Service domain endpoint is public.
      Source:
        Owner: AWS
        SourceIdentifier: ELASTICSEARCH_IN_VPC_ONLY
        MaximumExecutionFrequency:
          !FindInMap
          - Settings
            FrequencyMap
          - !Ref MaximumExecutionFrequency
```

Detect whether any Amazon EMR cluster master nodes have public IP addresses

This control detects whether any Amazon EMR cluster master nodes have public IP addresses.

This is a detective control with elective guidance. By default, this control isn't enabled on any OUs

**In the console:**

- The rule shows **Non-compliant** status if a master node has a public IP address.
- This control checks clusters that are in RUNNING or WAITING state.

The artifact for this control is the following AWS Config rule.
AWS Control Tower User Guide
Data residency controls

AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rule to detect whether any Amazon EMR cluster master nodes have public IP addresses

Parameters:
ConfigRuleName:
  Type: 'String'
  Description: 'Name for the Config rule'

MaximumExecutionFrequency:
  Type: String
  Default: 24hours
  Description: The frequency at which AWS Config will run evaluations for the rule.
  AllowedValues:
    - 1hour
    - 3hours
    - 6hours
    - 12hours
    - 24hours

Mappings:
  Settings:
    FrequencyMap:
      1hour : One_Hour
      3hours : Three_Hours
      6hours : Six_Hours
      12hours : Twelve_Hours
      24hours : TwentyFour_Hours

Resources:
EmrMasterNoPublicIp:
  Type: AWS::Config::ConfigRule
  Properties:
    ConfigRuleName: !Sub ${ConfigRuleName}
    Description: Detects whether any Amazon Elastic MapReduce (EMR) cluster master nodes have public IP addresses. The rule is NON_COMPLIANT if a master node has a public IP. This control checks clusters that are in RUNNING or WAITING state.
    Source:
      Owner: AWS
      SourceIdentifier: EMR_MASTER_NO_PUBLIC_IP
    MaximumExecutionFrequency: !FindInMap
      - Settings
      - FrequencyMap
      - !Ref MaximumExecutionFrequency

Detect whether the AWS Lambda function policy attached to the Lambda resource blocks public access

This control detects whether the AWS Lambda function policy attached to the Lambda resource blocks public access.

This is a detective control with elective guidance. By default, this control isn't enabled on any OUs.

In the console:

- The rule shows Non-compliant status if the Lambda function policy allows public access.

The artifact for this control is the following AWS Config rule.

AWSTemplateFormatVersion: 2010-09-09
Data residency controls

Description: Configure AWS Config rule to detect whether the AWS Lambda function policy attached to the Lambda resource blocks public access.

Parameters:
- ConfigRuleName:
  - Type: 'String'
  - Description: 'Name for the Config rule'

Resources:
- LambdaFunctionPublicAccessProhibited:
  - Type: AWS::Config::ConfigRule
  - Properties:
    - ConfigRuleName: !Sub ${ConfigRuleName}
    - Description: Detects whether the AWS Lambda function policy attached to the Lambda resource prohibits public access. The rule is NON_COMPLIANT if the Lambda function policy allows public access.
    - ComplianceResourceTypes:
      - AWS::Lambda::Function
    - Source:
      - Owner: AWS
      - SourceIdentifier: LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED

Detect whether public routes exist in the route table for an Internet Gateway (IGW)

This control detects whether public routes exist in the route table associated with an Internet Gateway (IGW).

This is a detective control with elective guidance. By default, this control isn't enabled on any OUs.

In the console:
- The rule shows Non-compliant status if a route has a destination CIDR block of 0.0.0.0/0 or ::/0 or if a destination CIDR block does not match the rule parameter.

  **Note**
  - This control fails if any of the routes to an IGW has a destination CIDR block of 0.0.0.0/0 or ::/0.

The artifact for this control is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rule to detect whether public routes exist in the route table for an Internet Gateway (IGW)

Parameters:
- ConfigRuleName:
  - Type: 'String'
  - Description: 'Name for the Config rule'

Resources:
- NoUnrestrictedRouteToIgw:
  - Type: AWS::Config::ConfigRule
  - Properties:
    - ConfigRuleName: !Sub ${ConfigRuleName}
    - Description: Detects whether public routes exist in the route table associated with an Internet Gateway (IGW). The rule is NON_COMPLIANT if a route has a destination CIDR block of '0.0.0.0/0' or '::/0' or if a destination CIDR block does not match the rule parameter.
    - Scope:
```

1356
Detect whether Amazon Redshift clusters are blocked from public access

This control detects whether Amazon Redshift clusters are blocked from public access.

This is a detective control with elective guidance. By default, this control isn't enabled on any OUs.

In the console:

• The rule shows **Non-compliant** status if the **publiclyAccessible** field is set to **True** in the cluster configuration item.

The artifact for this control is the following AWS Config rule.

```awsconfig
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rule to detect whether Amazon Redshift clusters are blocked from public access

Parameters:
ConfigRuleName:
  Type: 'String'
  Description: 'Name for the Config rule'

Resources:
RedshiftClusterPublicAccessCheck:
  Type: AWS::Config::ConfigRule
  Properties:
    ConfigRuleName: !Sub ${ConfigRuleName}
    Description: Detects whether Amazon Redshift clusters are blocked from public access. The rule is NON_COMPLIANT if the publiclyAccessible field is true in the cluster configuration item.
    Scope:
      ComplianceResourceTypes:
        - AWS::Redshift::Cluster
      Source:
        Owner: AWS
        SourceIdentifier: REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK
```

Detect whether an Amazon SageMaker notebook instance allows direct internet access

This control detects whether an Amazon SageMaker notebook instance allows direct internet access.

This is a detective control with elective guidance. By default, this control isn't enabled on any OUs.

In the console:

• The rule shows **Non-compliant** status if Amazon SageMaker notebook instances allow direct internet access.

The artifact for this control is the following AWS Config rule.
Detect whether any Amazon VPC subnets are assigned a public IP address

This control detects whether Amazon Virtual Private Cloud (Amazon VPC) subnets are assigned a public IP address.

This is a detective control with elective guidance. By default, this control isn't enabled on any OUs.

In the console:

- The rule shows Non-compliant status if the Amazon VPC has subnets that are assigned a public IP address.

The artifact for this control is the following AWS Config rule.
Data residency controls

**AWS Template Format Version: 2010-09-09**

**Description:** Detect whether any Amazon VPC subnets are assigned a public IP address

**Parameters:**
- **ConfigRuleName:**
  - Type: `String`
  - Description: 'Name for the Config rule'

**Resources:**
- **SubnetAutoAssignPublicIpDisabled:**
  - Type: `AWS::Config::ConfigRule`
    - Properties:
      - **ConfigRuleName:** !Sub ${ConfigRuleName}
        - Description: Detects whether Amazon Virtual Private Cloud (Amazon VPC) subnets are assigned a public IP address. The rule is NON_COMPLIANT if Amazon VPC has subnets that are assigned a public IP address.
        - **Scope:**
          - **ComplianceResourceTypes:**
            - AWS::EC2::Subnet
          - **Source:**
            - Owner: AWS
            - **SourceIdentifier:** SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

**Detect whether AWS Systems Manager documents owned by the account are public**

This control detects whether AWS Systems Manager documents owned by the account are public.

This is a detective control with elective guidance. By default, this control isn't enabled on any OUs.

In the console: The rule shows **Non-compliant** status if any documents with owner 'Self' are public.

The artifact for this control is the following AWS Config rule.

**AWS Template Format Version: 2010-09-09**

**Description:** Configure AWS Config rule to detect whether AWS Systems Manager documents owned by the account are public

**Parameters:**
- **ConfigRuleName:**
  - Type: `String`
  - Description: 'Name for the Config rule'

**MaximumExecutionFrequency:**
- **Type:** String
- **Default:** 24hours
  - Description: The frequency at which AWS Config will run evaluations for the rule.
  - **AllowedValues:**
    - 1hour
    - 3hours
    - 6hours
    - 12hours
    - 24hours

**Mappings:**
- **Settings:**
  - **FrequencyMap:**
    - 1hour : One_Hour
    - 3hours : Three_Hours
    - 6hours : Six_Hours
Optional controls

Optional controls in AWS Control Tower are applied at the OU level. You can activate and deactivate these optional controls through the AWS Control Tower console, or by means of the control APIs.

AWS Control Tower offers several types of optional controls:

- **Proactive controls (p. 230)**, which are based on AWS CloudFormation hooks.
- Security Hub controls, which are based on AWS Config rules – these controls are owned by Security Hub and integrated with AWS Control Tower, by means of the **Service-Managed Standard: AWS Control Tower**.
- Data residency controls, which are elective controls based on SCPs and AWS Config rules, implemented within AWS Control Tower.
- Strongly recommended controls, which are based on SCPs and AWS Config rules, implemented within AWS Control Tower.
- Elective controls, which are based on SCPs and AWS Config rules, implemented within AWS Control Tower.

The strongly recommended and elective controls owned by AWS Control Tower are optional, which means that you can customize the level of enforcement for OUs in your landing zone by choosing which ones to enable. Optional controls are not enabled by default. For more information about optional controls, see the following control reference pages in the next sections.

**Note**

It is important to know that some detective controls in AWS Control Tower do not operate in certain AWS Regions where AWS Control Tower is available, because those Regions do not support the required underlying functionality. As a result, when you deploy a detective control, the control may not be operating in all Regions that you govern with AWS Control Tower. For details, see Control limitations (p. 28).

You can view the Regions for each control in the AWS Control Tower console. For more information about the detective controls that cannot be deployed in certain Regions, see the Regional services list documentation to learn more about the Regions where AWS Config is available. If the detective control is implemented as a managed AWS Config rule, see the Security Hub controls reference documentation.

**Strongly recommended controls**

Strongly recommended controls are owned by AWS Control Tower. They are based on best practices for well-architected multi-account environments. These controls are not enabled by default, and they can be
deactivated through the AWS Control Tower console or the control APIs. Following, you'll find a reference for each of the strongly recommended controls available in AWS Control Tower.

Topics

- Disallow Creation of Access Keys for the Root User (p. 1361)
- Disallow Actions as a Root User (p. 1361)
- Detect Whether Encryption is Enabled for Amazon EBS Volumes Attached to Amazon EC2 Instances (p. 1362)
- Detect Whether Unrestricted Incoming TCP Traffic is Allowed (p. 1362)
- Detect Whether Unrestricted Internet Connection Through SSH is Allowed (p. 1364)
- Detect Whether MFA for the Root User is Enabled (p. 1365)
- Detect Whether Public Read Access to Amazon S3 Buckets is Allowed (p. 1365)
- Detect Whether Public Write Access to Amazon S3 Buckets is Allowed (p. 1366)
- Detect Whether Amazon EBS Volumes are Attached to Amazon EC2 Instances (p. 1366)
- Detect Whether Amazon EBS Optimization is Enabled for Amazon EC2 Instances (p. 1367)
- Detect Whether Public Access to Amazon RDS Database Instances is Enabled (p. 1368)
- Detect Whether Public Access to Amazon RDS Database Snapshots is Enabled (p. 1368)
- Detect Whether Storage Encryption is Enabled for Amazon RDS Database Instances (p. 1369)
- Detect whether an account has AWS CloudTrail or CloudTrail Lake enabled (p. 1369)

Disallow Creation of Access Keys for the Root User

Secures your AWS accounts by disallowing creation of access keys for the root user. We recommend that you instead create access keys for the IAM users or IAM Identity Center users, which grant limited permissions to interact with your AWS account. This is a preventive control with strongly recommended guidance. By default, this control is not enabled.

The artifact for this control is the following SCP.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRRESTRICTROOTUSERACCESSKEYS",
      "Effect": "Deny",
      "Action": "iam:CreateAccessKey",
      "Resource": ["*"]
    },
    "Condition": {
      "StringLike": {
        "aws:PrincipalArn": [
          "arn:aws:iam::*:root"
        ]
      }
    }
  ]
}
```

Disallow Actions as a Root User

Secures your AWS accounts by disallowing account access with root user credentials, which are credentials of the account owner that allow unrestricted access to all resources in the account. Instead,
we recommend that you create IAM Identity Center users for everyday interaction with your AWS account. This is a preventive control with strongly recommended guidance. By default, this control is not enabled.

The artifact for this control is the following SCP.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRRESTRICTROOTUSER",
            "Effect": "Deny",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:PrincipalArn": ["arn:aws:iam::*:root"]
                }
            }
        }
    ]
}
```

Detect Whether Encryption is Enabled for Amazon EBS Volumes Attached to Amazon EC2 Instances

This control detects whether the Amazon EBS volumes attached to an Amazon EC2 instance are encrypted. This control does not change the status of the account. This is a detective control with strongly recommended guidance. By default, this control isn't enabled on any OUs.

The artifact for this control is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check for encryption of all storage volumes attached to compute
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForEncryptedVolumes:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether EBS volumes that are in an attached state are encrypted.
      Source:
        Owner: AWS
        SourceIdentifier: ENCRYPTED_VOLUMES
      Scope:
        ComplianceResourceTypes:
        - AWS::EC2::Volume
```

Detect Whether Unrestricted Incoming TCP Traffic is Allowed

This control helps reduce a server's exposure to risk by detecting whether unrestricted incoming TCP traffic is allowed. It detects whether internet connections are enabled to Amazon EC2 instances through
services such as Remote Desktop Protocol (RDP). This control does not change the status of the account. This is a detective control with strongly recommended guidance. By default, this control is not enabled.

**Note**
This control fails if any of the rules in a security group allow ingress traffic from `0.0.0.0/0` or `::/0` for those ports.

The artifact for this control is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether security groups that are in use disallow unrestricted incoming TCP traffic to the specified ports.
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
  blockedPort1:
    Type: String
    Default: '20'
    Description: Blocked TCP port number.
  blockedPort2:
    Type: String
    Default: '21'
    Description: Blocked TCP port number.
  blockedPort3:
    Type: String
    Default: '3389'
    Description: Blocked TCP port number.
  blockedPort4:
    Type: String
    Default: '3306'
    Description: Blocked TCP port number.
  blockedPort5:
    Type: String
    Default: '4333'
    Description: Blocked TCP port number.
Conditions:
  blockedPort1:
    Fn::Not:
    - Fn::Equals:
      - ''
      - Ref: blockedPort1
  blockedPort2:
    Fn::Not:
    - Fn::Equals:
      - ''
      - Ref: blockedPort2
  blockedPort3:
    Fn::Not:
    - Fn::Equals:
      - ''
      - Ref: blockedPort3
  blockedPort4:
    Fn::Not:
    - Fn::Equals:
      - ''
      - Ref: blockedPort4
  blockedPort5:
    Fn::Not:
    - Fn::Equals:
      - ''
      - Ref: blockedPort5
Resources:
  CheckForRestrictedCommonPortsPolicy:
    Type: AWS::Config::ConfigRule
```
Properties:
    ConfigRuleName: !Sub ${ConfigRuleName}
    Description: Checks whether security groups that are in use disallow unrestricted incoming TCP traffic to the specified ports.
InputParameters:
    blockedPort1:
        Fn::If:
            - blockedPort1
            - Ref: blockedPort1
            - Ref: AWS::NoValue
    blockedPort2:
        Fn::If:
            - blockedPort2
            - Ref: blockedPort2
            - Ref: AWS::NoValue
    blockedPort3:
        Fn::If:
            - blockedPort3
            - Ref: blockedPort3
            - Ref: AWS::NoValue
    blockedPort4:
        Fn::If:
            - blockedPort4
            - Ref: blockedPort4
            - Ref: AWS::NoValue
    blockedPort5:
        Fn::If:
            - blockedPort5
            - Ref: blockedPort5
            - Ref: AWS::NoValue
Scope:
    ComplianceResourceTypes:
        - AWS::EC2::SecurityGroup
Source:
    Owner: AWS
    SourceIdentifier: RESTRICTED_INCOMING_TRAFFIC

Detect Whether Unrestricted Internet Connection Through SSH is Allowed

This control detects whether internet connections are allowed through remote services such as the Secure Shell (SSH) protocol. This control does not change the status of the account. This is a detective control with strongly recommended guidance. By default, this control is not enabled.

**Note**
This control fails if any of the rules in a security group allow ingress traffic from 0.0.0.0/0 or ::/0 for SSH traffic.

The artifact for this control is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether security groups that are in use disallow SSH
Parameters:
    ConfigRuleName:
        Type: 'String'
        Description: 'Name for the Config rule'
Resources:
    CheckForRestrictedSshPolicy:
        Type: AWS::Config::ConfigRule
        Properties:
            ConfigRuleName: !Sub ${ConfigRuleName}
            Description: Checks whether security groups that are in use disallow unrestricted incoming SSH traffic.
```
Optional controls

Scope:
ComplianceResourceTypes:
- AWS::EC2::SecurityGroup
Source:
Owner: AWS
SourceIdentifier: INCOMING_SSH_DISABLED

Detect Whether MFA for the Root User is Enabled

This control detects whether multi-factor authentication (MFA) is enabled for the root user of the management account. MFA reduces vulnerability risks from weak authentication by requiring an additional authentication code after the user name and password are successful. This control does not change the status of the account. This is a detective control with strongly recommended guidance. By default, this control is not enabled.

The artifact for this control is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to require MFA for root access to accounts
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
MaximumExecutionFrequency:
  Type: String
  Default: 24hours
  Description: The frequency that you want AWS Config to run evaluations for the rule.
  AllowedValues:
    - 1hour
    - 3hours
    - 6hours
    - 12hours
    - 24hours
Mappings:
  Settings:
    FrequencyMap:
      1hour : One_Hour
      3hours : Three_Hours
      6hours : Six_Hours
      12hours : Twelve_Hours
      24hours : TwentyFour_Hours
Resources:
  CheckForRootMfa:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether the root user of your AWS account requires multi-factor authentication for console sign-in.
      Source:
        Owner: AWS
        SourceIdentifier: ROOT_ACCOUNT_MFA_ENABLED
      MaximumExecutionFrequency:
        !FindInMap
        - Settings
        - FrequencyMap
        - !Ref MaximumExecutionFrequency
```

Detect Whether Public Read Access to Amazon S3 Buckets is Allowed

This control detects whether public read access is allowed to Amazon S3 buckets. It helps you maintain secure access to data stored in the buckets. This control does not change the status of the account. This is a detective control with strongly recommended guidance. By default, this control is not enabled.
The artifact for this control is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check that your S3 buckets do not allow public access.
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForS3PublicRead:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks that your S3 buckets do not allow public read access. If an S3 bucket policy or bucket ACL allows public read access, the bucket is noncompliant.
      Source:
        Owner: AWS
        SourceIdentifier: S3_BUCKET_PUBLIC_READ_PROHIBITED
      Scope:
        ComplianceResourceTypes:
          - AWS::S3::Bucket
```

Detect Whether Public Write Access to Amazon S3 Buckets is Allowed

This control detects whether public write access is allowed to Amazon S3 buckets. It helps you maintain secure access to data stored in the buckets. This control does not change the status of the account. This is a detective control with strongly recommended guidance. By default, this control is not enabled.

The artifact for this control is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check that your S3 buckets do not allow public access.
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForS3PublicWrite:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks that your S3 buckets do not allow public write access. If an S3 bucket policy or bucket ACL allows public write access, the bucket is noncompliant.
      Source:
        Owner: AWS
        SourceIdentifier: S3_BUCKET_PUBLIC_WRITE_PROHIBITED
      Scope:
        ComplianceResourceTypes:
          - AWS::S3::Bucket
```

Detect Whether Amazon EBS Volumes are Attached to Amazon EC2 Instances

This control detects whether an Amazon EBS volume device persists independently from an Amazon EC2 instance. This control does not change the status of the account. This is a detective control with strongly recommended guidance. By default, this control is not enabled.

The artifact for this control is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
```
Description: Configure AWS Config rules to check whether EBS volumes are attached to EC2 instances

Parameters:
- ConfigRuleName:
  Type: 'String'
  Description: 'Name for the Config rule'
- deleteOnTermination:
  Type: 'String'
  Default: 'None'
  Description: 'Check for Delete on termination'

Conditions:
- deleteOnTermination:
  Fn::Not:
  - Fn::Equals:
    - 'None'
    - Ref: deleteOnTermination

Resources:
- CheckForEc2VolumesInUse:
  Type: AWS::Config::ConfigRule
  Properties:
    ConfigRuleName: !Sub ${ConfigRuleName}
    Description: Checks whether EBS volumes are attached to EC2 instances
    InputParameters:
      deleteOnTermination:
        Fn::If:
          - deleteOnTermination
          - Ref: deleteOnTermination
          - Ref: AWS::NoValue
    Source:
      Owner: AWS
      SourceIdentifier: EC2_VOLUME_INUSE_CHECK
    Scope:
      ComplianceResourceTypes:
      - AWS::EC2::Volume

Detect Whether Amazon EBS Optimization isEnabled for Amazon EC2 Instances

Detects whether Amazon EC2 instances are launched without an Amazon EBS volume that is optimized for performance. Amazon EBS-optimized volumes minimize contention between Amazon EBS I/O and other traffic from your instance. This control does not change the status of the account. This is a detective control with strongly recommended guidance. By default, this control is not enabled.

The artifact for this control is the following AWS Config rule.

AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether EBS optimization is enabled for your EC2 instances that can be EBS-optimized
Parameters:
- ConfigRuleName:
  Type: 'String'
  Description: 'Name for the Config rule'

Resources:
- CheckForEbsOptimizedInstance:
  Type: AWS::Config::ConfigRule
  Properties:
    ConfigRuleName: !Sub ${ConfigRuleName}
    Description: Checks whether EBS optimization is enabled for your EC2 instances that can be EBS-optimized
    Source:
      Owner: AWS
      SourceIdentifier: EBS_OPTIMIZED_INSTANCE
    Scope:
      ComplianceResourceTypes:
- AWS::EC2::Instance

**Detect Whether Public Access to Amazon RDS Database Instances is Enabled**

Detects whether your Amazon RDS database instances allow public access. You can secure your Amazon RDS database instances by disallowing public access. This control does not change the status of the account. This is a detective control with strongly recommended guidance. By default, this control is not enabled.

The artifact for this control is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether Amazon RDS instances are not publicly accessible.
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForRdsPublicAccess:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub '${ConfigRuleName}'}
      Description: Checks whether the Amazon Relational Database Service (RDS) instances are not publicly accessible. The rule is non-compliant if the publiclyAccessible field is true in the instance configuration item.
      Source:
        Owner: AWS
        SourceIdentifier: RDS_INSTANCE_PUBLIC_ACCESS_CHECK
      Scope:
        ComplianceResourceTypes:
          - AWS::RDS::DBInstance
```

**Detect Whether Public Access to Amazon RDS Database Snapshots is Enabled**

Detects whether your Amazon RDS database snapshots have public access enabled. You can protect your information by disabling public access. This control does not change the status of the account. This is a detective control with strongly recommended guidance. By default, this control is not enabled.

The artifact for this control is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Checks if Amazon Relational Database Service (Amazon RDS) snapshots are public.
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForRdsStorageEncryption:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub '${ConfigRuleName}'}
      Description: Checks if Amazon Relational Database Service (Amazon RDS) snapshots are public. The rule is non-compliant if any existing and new Amazon RDS snapshots are public.
      Source:
        Owner: AWS
        SourceIdentifier: RDS_SNAPSHOTS_PUBLIC_PROHIBITED
      Scope:
        ComplianceResourceTypes:
          - AWS::RDS::DBSnapshot
```
Detect Whether Storage Encryption is Enabled for Amazon RDS Database Instances

Detects Amazon RDS database instances that are not encrypted at rest. You can secure your Amazon RDS database instances at rest by encrypting the underlying storage for database instances and their automated backups, Read Replicas, and snapshots. This control does not change the status of the account. This is a detective control with strongly recommended guidance. By default, this control is not enabled.

The artifact for this control is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether storage encryption is enabled for your RDS DB instances
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForRdsStorageEncryption:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether storage encryption is enabled for your RDS DB instances.
      Source:
        Owner: AWS
        SourceIdentifier: RDS_STORAGE_ENCRYPTED
      Scope:
        ComplianceResourceTypes:
        - AWS::RDS::DBInstance
```

Detect whether an account has AWS CloudTrail or CloudTrail Lake enabled

This control detects whether an account has AWS CloudTrail or CloudTrail Lake enabled. The rule is NON_COMPLIANT if either CloudTrail or CloudTrail Lake is not enabled in an account. This is a detective control with strongly recommended guidance. By default, this control is not enabled on any OUs.

The artifact for this control is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to detect whether an account has AWS CloudTrail or CloudTrail Lake enabled.
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForCloudtrailEnabled:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Detects whether an account has AWS CloudTrail or CloudTrail Lake enabled. The rule is NON_COMPLIANT if either CloudTrail or CloudTrail Lake is not enabled in an account.
      Source:
        Owner: AWS
        SourceIdentifier: CLOUD_TRAIL_ENABLED
```

1369
Elective controls

Elective controls enable you to lock down or track attempts at performing commonly restricted actions in an AWS enterprise environment. These controls are not enabled by default, and can be disabled. Following, you'll find a reference for the elective controls available in AWS Control Tower. The elective controls specifically for data residency are collected into a separate section, Controls that enhance data residency protection (p. 1343).

Topics

- Disallow Changes to Encryption Configuration for Amazon S3 Buckets [Previously: Enable Encryption at Rest for Log Archive] (p. 1370)
- Disallow Changes to Logging Configuration for Amazon S3 Buckets [Previously: Enable Access Logging for Log Archive] (p. 1371)
- Disallow Changes to Bucket Policy for Amazon S3 Buckets [Previously: Disallow Policy Changes to Log Archive] (p. 1371)
- Disallow Changes to Lifecycle Configuration for Amazon S3 Buckets [Previously: Set a Retention Policy for Log Archive] (p. 1371)
- Disallow Changes to Replication Configuration for Amazon S3 Buckets (p. 1372)
- Disallow Delete Actions on Amazon S3 Buckets Without MFA (p. 1372)
- Detect Whether MFA is Enabled for AWS IAM Users (p. 1372)
- Detect Whether MFA is Enabled for AWS IAM Users of the AWS Console (p. 1374)
- Detect Whether Versioning for Amazon S3 Buckets is Enabled (p. 1374)
- Disallow management of resource types, modules, and hooks within the AWS CloudFormation registry (p. 1375)

Disallow Changes to Encryption Configuration for Amazon S3 Buckets [Previously: Enable Encryption at Rest for Log Archive]

This control disallows changes to encryption for all Amazon S3 buckets. This is a preventive control with elective guidance. By default, this control is not enabled.

The artifact for this control is the following service control policy (SCP).

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRAUDITBUCKETENCRYPTIONENABLED",
         "Effect": "Deny",
         "Action": [
            "s3:PutEncryptionConfiguration"
         ],
         "Resource": ["*"],
         "Condition": {
            "ArnNotLike": {
               "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
            }
         }
      }
   ]
}
```
Disallow Changes to Logging Configuration for Amazon S3 Buckets [Previously: Enable Access Logging for Log Archive]

This control disallows changes to logging configuration for all Amazon S3 buckets. This is a preventive control with elective guidance. By default, this control is not enabled.

The artifact for this control is the following SCP.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRAUDITBUCKETLOGGINGENABLED",
      "Effect": "Deny",
      "Action": ["s3:PutBucketLogging"],
      "Resource": ["*"],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
  ]
}
```

Disallow Changes to Bucket Policy for Amazon S3 Buckets [Previously: Disallow Policy Changes to Log Archive]

This control disallows changes to bucket policy for all Amazon S3 buckets. This is a preventive control with elective guidance. By default, this control is not enabled.

The artifact for this control is the following SCP.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRAUDITBUCKETPOLICYCHANGESPROHIBITED",
      "Effect": "Deny",
      "Action": ["s3:PutBucketPolicy"],
      "Resource": ["*"],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
  ]
}
```

Disallow Changes to Lifecycle Configuration for Amazon S3 Buckets [Previously: Set a Retention Policy for Log Archive]

This control disallows lifecycle configuration changes for all Amazon S3 buckets. This is a preventive control with elective guidance. By default, this control is not enabled.
The artifact for this control is the following SCP.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRAUDITBUCKETRETENTIONPOLICY",
         "Effect": "Deny",
         "Action": [
            "s3:PutLifecycleConfiguration"
         ],
         "Resource": ["*"]
      }
   ]
}
```

Disallow Changes to Replication Configuration for Amazon S3 Buckets

Prevents changes to the way your Amazon S3 buckets have been set up to handle replication within Regions or across Regions. For example, if you set up your buckets with single-region replication, to restrict the location of your Amazon S3 data to a single AWS Region (thereby disabling any automatic, asynchronous copying of objects across buckets to other AWS Regions), then this control prevents that replication setting from being changed. This is a preventive control with elective guidance. By default, this control is not enabled.

The artifact for this control is the following SCP.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRRESTRICTS3CROSSREGIONREPLICATION",
         "Effect": "Deny",
         "Action": [
            "s3:PutReplicationConfiguration"
         ],
         "Resource": ["*"]
      }
   ]
}
```

Disallow Delete Actions on Amazon S3 Buckets Without MFA

Protects your Amazon S3 buckets by requiring MFA for delete actions. MFA requires an extra authentication code after the user name and password are successful. This is a preventive control with elective guidance. By default, this control is not enabled.

The artifact for this control is the following SCP.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRREQUIREMAFADELETE",
         "Effect": "Deny",
         "Action": [
            "s3:DeleteObject"
         ],
         "Resource": ["*"]
      }
   ]
}
```
Detect Whether MFA is Enabled for AWS IAM Users

This control detects whether MFA is enabled for AWS IAM users. You can protect your account by requiring MFA for all AWS users in the account. MFA requires an additional authentication code after the user name and password are successful. This control does not change the status of the account. This is a detective control with elective guidance. By default, this control is not enabled.

The artifact for this control is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether the IAM users have MFA enabled
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
  MaximumExecutionFrequency:
    Type: String
    Default: 1hour
    Description: The frequency that you want AWS Config to run evaluations for the rule.
    AllowedValues:
      - 1hour
      - 3hours
      - 6hours
      - 12hours
      - 24hours
Mappings:
  Settings:
    FrequencyMap:
      1hour : One_Hour
      3hours : Three_Hours
      6hours : Six_Hours
      12hours : Twelve_Hours
      24hours : TwentyFour_Hours
Resources:
  CheckForIAMUserMFA:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether the AWS Identity and Access Management users have multi-factor authentication (MFA) enabled. The rule is COMPLIANT if MFA is enabled.
      Source:
        Owner: AWS
        SourceIdentifier: IAM_USER_MFA_ENABLED
      MaximumExecutionFrequency:
```
Detect Whether MFA is Enabled for AWS IAM Users of the AWS Console

Protects your account by requiring MFA for all AWS IAM users in the console. MFA reduces vulnerability risks from weak authentication by requiring an additional authentication code after the user name and password are successful. This control detects whether MFA is enabled. This control does not change the status of the account. This is a detective control with elective guidance. By default, this control is not enabled.

The artifact for this control is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether MFA is enabled for all AWS IAM users that use a console password.
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
  MaximumExecutionFrequency:
    Type: String
    Default: 1hour
    Description: The frequency that you want AWS Config to run evaluations for the rule.
    AllowedValues:
      - 1hour
      - 3hours
      - 6hours
      - 12hours
      - 24hours
Mappings:
  Settings:
    FrequencyMap:
      1hour   : One_Hour
      3hours  : Three_Hours
      6hours  : Six_Hours
      12hours : Twelve_Hours
      24hours : TwentyFour_Hours
Resources:
  CheckForIAMUserConsoleMFA:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether AWS Multi-Factor Authentication (MFA) is enabled for all AWS Identity and Access Management (IAM) users that use a console password. The rule is COMPLIANT if MFA is enabled.
      Source:
        Owner: AWS
        SourceIdentifier: MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS
      MaximumExecutionFrequency:
        !FindInMap
        - Settings
        - FrequencyMap
        - !Ref MaximumExecutionFrequency
```

Detect Whether Versioning for Amazon S3 Buckets is Enabled

Detects whether your Amazon S3 buckets are enabled for versioning. Versioning allows you to recover objects from accidental deletion or overwrite. This control does not change the status of the account. This is a detective control with elective guidance. By default, this control is not enabled.
The artifact for this control is the following AWS Config rule.

```json
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether versioning is enabled for your S3 buckets.
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForS3VersioningEnabled:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether versioning is enabled for your S3 buckets.
      Source:
        Owner: AWS
        SourceIdentifier: S3_BUCKET_VERSIONING_ENABLED
      Scope:
        ComplianceResourceTypes:
          - AWS::S3::Bucket

Disallow management of resource types, modules, and hooks within the AWS CloudFormation registry

This elective control disallows management of the following extension types in the AWS CloudFormation registry: resource types, modules, and hooks. For more information about AWS CloudFormation extensions, see Using the AWS CloudFormation registry.

Note
You must enable this control when you activate proactive controls in your environment.

- **Control objective**: Protect configurations
- **Implementation** Service control policy (SCP)
- **Control behavior**: Preventive
- **Control guidance**: Elective
- **Control owner**: AWS Control Tower
- **Control ID**: CT.CLOUDFORMATION.PR.1
- **Severity**: Critical
- **AWS Service**: AWS CloudFormation
- **Resource types**: AWS::CloudFormation::HookDefaultVersion, AWS::CloudFormation::HookTypeConfig, AWS::CloudFormation::HookVersion, AWS::CloudFormation::ModuleDefaultVersion, AWS::CloudFormation::ModuleVersion, AWS::CloudFormation::ResourceDefaultVersion, AWS::CloudFormation::ResourceVersion

The following example shows the SCP artifact for this control.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRDISALLOWMODIFICATIONCFNREGISTRY",
      "Effect": "Deny",
      "Action": [
        "cloudformation:RegisterType",
```
Optional controls

- `cloudformation:DeregisterType`
- `cloudformation:SetTypeConfiguration`
- `cloudformation:SetTypeDefaultVersion`
- `cloudformation:PublishType`

```
"Resource": [ "*
"Condition": {
  "ArnNotLike": {
    "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
  }
}
```


AWS Control Tower is a service that's built on top of other AWS services, to assist you in setting up a well-architected environment. This chapter provides a brief overview of these services, including configuration information about the underlying services and how they work in AWS Control Tower.

For more information about how to measure a well-architected environment, learn about the AWS Well-Architected Tool. Also see the Management and Governance Cloud Environment Guide.

**Topics**

- Deploy Environments with AWS CloudFormation (p. 1377)
- Monitor Events with CloudTrail (p. 1377)
- Monitor Resources and Services with CloudWatch (p. 1378)
- Govern Resource Configurations with AWS Config (p. 1378)
- AWS Key Management Service (p. 1379)
- Run Serverless Compute Functions with Lambda (p. 1379)
- Manage Accounts Through AWS Organizations (p. 1379)
- Store Objects with Amazon S3 (p. 1380)
- Monitor your environment with Security Hub (p. 1380)
- Provision Accounts Through Service Catalog (p. 1380)
- Manage Users and Access Through AWS IAM Identity Center (p. 1380)
- Track Alerts Through Amazon Simple Notification Service (p. 1384)
- Build Distributed Applications with AWS Step Functions (p. 1385)

### Deploy Environments with AWS CloudFormation

AWS CloudFormation enables you to create and provision AWS infrastructure deployments predictably and repeatedly. It helps you leverage AWS products to build highly reliable, highly scalable, cost-effective applications in the cloud without worrying about creating and configuring the underlying AWS infrastructure. AWS CloudFormation enables you to use a template file to create and delete a collection of resources together as a single unit (a stack). For more information, see AWS CloudFormation User Guide.

AWS Control Tower uses AWS CloudFormation stacksets to apply controls on accounts. For more information about how AWS CloudFormation and AWS Control Tower work together, see Creating AWS Control Tower resources with AWS CloudFormation (p. 54).

### Monitor Events with CloudTrail

AWS Control Tower configures AWS CloudTrail to enable centralized logging and auditing. With CloudTrail, the management account can review administrative actions and lifecycle events for member accounts.
CloudTrail helps you monitor your AWS environment in the cloud by keeping a history of AWS API calls for your accounts. For example, you can identify the users and accounts that called AWS APIs for services that support CloudTrail, the source IP address from which the calls were made, and the time when the calls occurred. You can integrate CloudTrail into applications using the API, automate trail creation for your organization, check the status of your trails, and control how administrators turn CloudTrail logging on and off. For more information, see AWS CloudTrail User Guide.

Organization-level trails

AWS Control Tower sets up a new CloudTrail trail when you set up a landing zone. It is an organization-level trail, which means that it logs all events for the management account and all member accounts in the organization. This feature relies on trusted access to give the management account permissions to create a trail on every member account.

For more information about AWS Control Tower and CloudTrail organization trails, see Creating a trail for an organization.

Note
In AWS Control Tower releases before landing zone version 3.0, AWS Control Tower created a member account trail in each account. When you update to release 3.0, your CloudTrail trail becomes an organization trail. For best practices when moving between trails, see Best practices for changing trails in the CloudTrail User Guide.

When you enroll an account into AWS Control Tower, your account is governed by the AWS CloudTrail trail for the AWS Control Tower organization. If you have an existing deployment of a CloudTrail trail in that account, you may see duplicate charges unless you delete the existing trail for the account before you enroll it in AWS Control Tower.

Note
When you update to landing zone version 3.0, AWS Control Tower deletes the account-level trails of your enrolled accounts on your behalf. Your existing, account-level log files are preserved in their Amazon S3 bucket.

Monitor Resources and Services with CloudWatch

Amazon CloudWatch provides a reliable, scalable, and flexible monitoring solution that you can start using within minutes. You no longer need to set up, manage, and scale your own monitoring systems and infrastructure. For more information, see Amazon CloudWatch User Guide.

For more information about how Amazon CloudWatch works with AWS Control Tower, see Monitoring.

Govern Resource Configurations with AWS Config

AWS Config provides a detailed view of the resources associated with your AWS account, including how they are configured, how they are related to one another, and how the configurations and their relationships have changed over time. For more information, see AWS Config Developer Guide.

AWS Config resources provisioned by AWS Control Tower are tagged automatically with aws-control-tower and a value of managed-by-control-tower.

For more information about how AWS Config monitors and records resources in AWS Control Tower, and how it bills you for them, see Monitoring resource changes with AWS Config (p. 1413).

AWS Control Tower uses AWS Config Rules to implement detective controls. For more information, see About controls in AWS Control Tower (p. 196).
AWS Key Management Service

AWS Key Management Service allows you to create and control keys that protect your data. AWS Control Tower optionally allows you to encrypt your data with AWS KMS encryption keys. For information about AWS KMS, see the AWS KMS Developer Guide.

For information about how to set up AWS KMS keys with AWS Control Tower, see Step 2. Configure and launch your landing zone (p. 20).

Run Serverless Compute Functions with Lambda

With AWS Lambda, you can run code without provisioning or managing servers. You can run code for many types of application or backend service—with no need for additional administration overhead. When you upload your code, Lambda can run and scale the code with high availability. You can set up your code to trigger from other AWS services automatically, or you can call it directly from any web or mobile app.

For example, certain roles in the AWS Control Tower audit account can be assumed programmatically, so that you can review other accounts using Lambda. Also, you can use AWS Control Tower lifecycle events to trigger Lambda functions.

Manage Accounts Through AWS Organizations

AWS Organizations is an account management service that lets you consolidate multiple AWS accounts into an organization that you create and centrally manage. With Organizations, you can create member accounts and invite existing accounts to join your organization. You can organize those accounts into groups and attach policy-based controls. For more information, see AWS Organizations User Guide.

In AWS Control Tower, Organizations helps centrally manage billing; control access, compliance, and security; and share resources across your member AWS accounts. Accounts are grouped into logical groups, called organizational units (OUs). For more information on Organizations, see AWS Organizations User Guide.

AWS Control Tower uses the following OUs:

- **Root** – The parent container for all accounts and all other OUs in your landing zone.
- **Security** – This OU contains the log archive account, the audit account, and the resources they own.
- **Sandbox** – This OU is created when you set up your landing zone. It and other child OUs in your landing zone contain your member accounts. These are the accounts that your end users access to perform work on AWS resources.

**Note**

You can add additional OUs in your landing zone through the AWS Control Tower console on the Organizational units page.

Considerations

OUs created through AWS Control Tower can have controls applied to them. OUs created outside of AWS Control Tower cannot, by default. You can, however, register such OUs. Once you have registered an
OU, you can apply controls to it and its accounts. For information on registering an OU, see Register an existing organizational unit with AWS Control Tower (p. 190).

Store Objects with Amazon S3

Amazon Simple Storage Service (Amazon S3) is storage for the internet. You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web. You can accomplish these tasks using the simple and intuitive web interface of the AWS Management Console. For more information, see Amazon Simple Storage Service User Guide.

When you set up your landing zone, an Amazon S3 bucket is created in your log archive account to contain all logs across all accounts in your landing zone.

Monitor your environment with Security Hub


Provision Accounts Through Service Catalog

Service Catalog enables IT administrators to create, manage, and distribute portfolios of approved products to end users, who then have access to the products they need in a personalized portal. Typical products include servers, databases, websites, or applications that are deployed using AWS resources.

You can control the users that have access to specific products, which allows you to enforce compliance with organizational business standards, manage product lifecycles, and help users find and launch products with confidence. For more information, see Service Catalog Administrator Guide.

In AWS Control Tower, your central cloud administrators and your end users can provision accounts in your landing zone using Account Factory, a product in Service Catalog. For more information, see Provision and manage accounts with Account Factory (p. 121).

AWS Control Tower also can make use of the Service Catalog APIs to further automate account provisioning and updating. For details, see the AWS Service Catalog Developer Guide.

Manage Users and Access Through AWS IAM Identity Center

AWS IAM Identity Center is a web-based service for securely controlling access to other AWS services. With IAM, you can centrally manage users, along with their security credentials—such as access keys and permissions—that designate the AWS resources to which specific users and applications are granted access.

AWS Identity and Access Management (IAM) simplifies how you manage access to AWS accounts and business applications. You can control IAM Identity Center access and user permissions across all your AWS accounts in AWS Control Tower.

With IAM Identity Center, you also can administer access to popular business applications and custom applications that support Security Assertion Markup Language (SAML) 2.0. Also, IAM Identity Center
offers a user portal where your users can find all their assigned AWS accounts, business applications, and custom applications in one place. For more information, see AWS IAM Identity Center User Guide.

Working With AWS IAM Identity Center and AWS Control Tower

In AWS Control Tower, IAM Identity Center allows central cloud administrators and end-users to manage access to multiple AWS accounts and business applications. By default, AWS Control Tower uses this service to set up and manage access to the accounts created through Account Factory, unless you have selected the option to self-manage your identity and access control.

For a brief tutorial about how to set up your IAM Identity Center users and permissions in AWS Control Tower, you can view this video (6:23). For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Setting Up AWS IAM Identity Center in AWS Control Tower.

About setting up AWS Control Tower with IAM Identity Center

When you initially set up AWS Control Tower, only the root user user and any IAM users with the correct permissions can add IAM Identity Center users. However, after end users have been added in the AWSAccountFactory group, they can create new IAM Identity Center users from the Account Factory wizard. For more information, see Provision and manage accounts with Account Factory (p. 121).

If you choose the recommended default, AWS Control Tower sets up your landing zone with a preconfigured directory that helps you manage user identities and single sign-on, so that your users have federated access across accounts. When you set up your landing zone, this default directory is created to contain user groups and permission sets.

Note
You can delegate administration of AWS IAM Identity Center in your organization to an account other than the management account, by using the delegated administrator feature of IAM Identity Center. If you choose to use this feature, be aware that Administrators with access to manage group membership also can manage groups assigned to the management account. For more information, see this blog post, entitled, Getting started with AWS SSO delegated administration

User Groups, Roles, and Permission Sets

User groups manage specialized roles that are defined within your shared accounts. Roles establish sets of permissions that belong together. All members of a group inherit the permission sets, or roles, associated with the group. You can create new groups for the end users of your member accounts, so that you can custom-assign only the roles that are needed for the specific tasks a group performs.

The permission sets available cover a broad range of distinct user permission requirements, such as read-only access, AWS Control Tower administrative access, and Service Catalog access. These permission sets enable your end users to provision their own AWS accounts in your landing zone quickly, and in compliance with your enterprise's guidelines.

For tips on planning your allocations of users, groups, and permissions, refer to Recommendations for setting up groups, roles, and policies (p. 40).

For more information on how to use this service in the context of AWS Control Tower, see the following topics in the AWS IAM Identity Center User Guide.

- To add users, see Add Users.
- To add users to groups, see Add Users to Groups.
- To edit user properties, see Edit User Properties.
- To add a group, see Add Groups.
**Warning**

AWS Control Tower sets up your IAM Identity Center directory in your home region. If you set up your landing zone in another Region and then navigate to the IAM Identity Center console, you must change the Region to your home region. Do not delete your IAM Identity Center configuration in your home region.

**Things to Know About IAM Identity Center Accounts and AWS Control Tower**

Here are some good things to know when working with IAM Identity Center user accounts in AWS Control Tower.

- If your AWS IAM Identity Center user account is disabled, you'll get an error message when trying to provision new accounts in Account Factory. You can re-enable your IAM Identity Center user in the IAM Identity Center console.

- If you specify a new IAM Identity Center user email address when you update the provisioned product associated with an account that was vended by Account Factory, AWS Control Tower creates a new IAM Identity Center user account. The previously created user account is not removed. If you prefer to remove the previous IAM Identity Center user email address from AWS IAM Identity Center, see **Disabling a User**.

- AWS IAM Identity Center has been integrated with Azure Active Directory, and you can connect your existing Azure Active Directory to AWS Control Tower.

- For more information about how the behavior of AWS Control Tower interacts with AWS IAM Identity Center and different identity sources, refer to the Considerations for Changing Your Identity Source in the AWS IAM Identity Center documentation.

**IAM Identity Center Groups for AWS Control Tower**

AWS Control Tower offers preconfigured groups to organize users that perform specific tasks in your accounts. You can add users and assign them to these groups directly in IAM Identity Center. Doing so matches permission sets to users in groups within your accounts. The following groups are created when you set up your landing zone.

**AWSAccountFactory**

<table>
<thead>
<tr>
<th>Account</th>
<th>Permission sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management account</td>
<td>AWSServiceCatalogEndUserAccess</td>
<td>This group is only used in this account to provision new accounts using Account Factory.</td>
</tr>
</tbody>
</table>

**AWSServiceCatalogAdmins**

<table>
<thead>
<tr>
<th>Account</th>
<th>Permission sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management account</td>
<td>AWSServiceCatalogAdminFullAccess</td>
<td>This group is only used in this account to make administrative changes to Account Factory. Users in this group can't provision new accounts unless they're also in the AWSAccountFactory group.</td>
</tr>
</tbody>
</table>
## AWSControlTowerAdmins

<table>
<thead>
<tr>
<th>Account</th>
<th>Permission sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management account</td>
<td>AWSAdministratorAccess</td>
<td>Users of this group in this account are the only ones that have access to the AWS Control Tower console.</td>
</tr>
<tr>
<td>Log archive account</td>
<td>AWSAdministratorAccess</td>
<td>Users have administrator access in this account.</td>
</tr>
<tr>
<td>Audit account</td>
<td>AWSAdministratorAccess</td>
<td>Users have administrator access in this account.</td>
</tr>
<tr>
<td>Member accounts</td>
<td>AWSOrganizationsFullAccess</td>
<td>Users have full access to Organizations in this account.</td>
</tr>
</tbody>
</table>

## AWSSecurityAuditPowerUsers

<table>
<thead>
<tr>
<th>Account</th>
<th>Permission sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management account</td>
<td>AWSPowerUserAccess</td>
<td>Users can perform application development tasks and can create and configure resources and services that support AWS aware application development.</td>
</tr>
<tr>
<td>Log archive account</td>
<td>AWSPowerUserAccess</td>
<td>Users can perform application development tasks and can create and configure resources and services that support AWS aware application development.</td>
</tr>
<tr>
<td>Audit account</td>
<td>AWSPowerUserAccess</td>
<td>Users can perform application development tasks and can create and configure resources and services that support AWS aware application development.</td>
</tr>
<tr>
<td>Member accounts</td>
<td>AWSPowerUserAccess</td>
<td>Users can perform application development tasks and can create and configure resources and services that support AWS aware application development.</td>
</tr>
</tbody>
</table>

## AWSSecurityAuditors

<table>
<thead>
<tr>
<th>Account</th>
<th>Permission sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management account</td>
<td>AWSReadOnlyAccess</td>
<td>Users have read-only access to all AWS services and resources in this account.</td>
</tr>
<tr>
<td>Log archive account</td>
<td>AWSReadOnlyAccess</td>
<td>Users have read-only access to all AWS services and resources in this account.</td>
</tr>
<tr>
<td>Account</td>
<td>Permission sets</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Audit account</td>
<td>AWSReadOnlyAccess</td>
<td>Users have read-only access to all AWS services and resources in this account.</td>
</tr>
<tr>
<td>Member accounts</td>
<td>AWSReadOnlyAccess</td>
<td>Users have read-only access to all AWS services and resources in this account.</td>
</tr>
</tbody>
</table>

**AWSLogArchiveAdmins**

<table>
<thead>
<tr>
<th>Account</th>
<th>Permission sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log archive account</td>
<td>AWSAdministratorAccess</td>
<td>Users have administrator access in this account.</td>
</tr>
</tbody>
</table>

**AWSLogArchiveViewers**

<table>
<thead>
<tr>
<th>Account</th>
<th>Permission sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log archive account</td>
<td>AWSReadOnlyAccess</td>
<td>Users have read-only access to all AWS services and resources in this account.</td>
</tr>
</tbody>
</table>

**AWSAuditAccountAdmins**

<table>
<thead>
<tr>
<th>Account</th>
<th>Permission sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit account</td>
<td>AWSAdministratorAccess</td>
<td>Users have administrator access in this account.</td>
</tr>
</tbody>
</table>

---

**Track Alerts Through Amazon Simple Notification Service**

Amazon Simple Notification Service (Amazon SNS) is a web service that enables applications, end-users, and devices to send and receive notifications instantly from the cloud. For more information, see the [Amazon Simple Notification Service Developer Guide](#).

AWS Control Tower uses Amazon SNS to send programmatic alerts to the email addresses of your management account and your audit account. These alerts help you prevent drift within your landing zone. For more information, see [Detect and resolve drift in AWS Control Tower](#) (p. 169).

We also use Amazon Simple Notification Service to send compliance notifications from AWS Config.

**Tip**

One of the best ways to receive AWS Control Tower control compliance notifications (in your audit account) is to subscribe to `AggregateConfigurationNotifications`. It is a service that helps you inspect compliance. It gives you real data about AWS Config rules going out of compliance. AWS Config automatically maintains the list of accounts in your OU.

You must subscribe manually, using email or any type of subscription that SNS allows. The statement `arn:aws:sns:homeregion:account:aws-controltower-AggregateSecurityNotifications` leads to your audit account.
Build Distributed Applications with AWS Step Functions

AWS Step Functions makes it easy to coordinate the components of distributed applications as a series of steps in a visual workflow. You can quickly build and run state machines to execute the steps of your application in a reliable and scalable fashion. For more information, see the AWS Step Functions Developer Guide.
Security in AWS Control Tower

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. The effectiveness of our security is regularly tested and verified by third-party auditors as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS Control Tower, see AWS Services in Scope by Compliance Program.

- **Security in the cloud** – Your responsibility is determined by the AWS services that you use. You are also responsible for other factors including the sensitivity of your data, your organization's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS Control Tower. The following topics show you how to configure AWS Control Tower to meet your security and compliance objectives. You also learn how to use other AWS services that help you monitor and secure your AWS Control Tower resources.

Data Protection in AWS Control Tower

The AWS shared responsibility model applies to data protection in AWS Control Tower. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the AWS Security Blog.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers’ email addresses, into tags or free-form text fields such as a Name field. This includes when you work with AWS Control Tower or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.
**Note**
User activity logging with AWS CloudTrail is handled automatically in AWS Control Tower when you set up your landing zone.

For more information about data protection, see the AWS Shared Responsibility Model and GDPR blog post on the AWS Security Blog. AWS Control Tower provides the following options that you can use to help secure the content that exists in your landing zone:

**Topics**
- Encryption at Rest (p. 1387)
- Encryption in Transit (p. 1387)
- Restrict Access to Content (p. 1387)

**Encryption at Rest**
AWS Control Tower uses Amazon S3 buckets and Amazon DynamoDB databases that are encrypted at rest by using Amazon S3-Managed Keys (SSE-S3) in support of your landing zone. This encryption is configured by default when you set up your landing zone. Optionally, you can configure your landing zone to encrypt resources with KMS encryption keys. You can also establish encryption at rest for the services you use in your landing zone for the services that support it. For more information, see the security chapter of that service's online documentation.

**Encryption in Transit**
AWS Control Tower uses Transport Layer Security (TLS) and client-side encryption for encryption in transit in support of your landing zone. In addition, accessing AWS Control Tower requires using the console, which can only be accessed through an HTTPS endpoint. This encryption is configured by default when you set up your landing zone.

**Restrict Access to Content**
As a best practice, you should restrict access to the appropriate subset of users. With AWS Control Tower, you can do this by ensuring that your central cloud administrators and end users have the right IAM permissions or, in the case of IAM Identity Center users, that they are in the correct groups.

- For more information about roles and policies for IAM entities, see IAM User Guide.
- For more information about the IAM Identity Center groups that are created when you set up your landing zone, see IAM Identity Center Groups for AWS Control Tower (p. 1382).

**Identity and access management in AWS Control Tower**
To perform any operation in your landing zone, such as provisioning accounts in Account Factory or creating new organizational units (OUs) in the AWS Control Tower console, either AWS Identity and Access Management (IAM) or AWS IAM Identity Center require you to authenticate that you're an approved AWS user. For example, if you're using the AWS Control Tower console, you authenticate your identity by providing your AWS credentials, as provided by your administrator.

After you authenticate your identity, IAM controls your access to AWS with a defined set of permissions on a specific set of operations and resources. If you are an account administrator, you can use IAM to control the access of other IAM users to the resources that are associated with your account.
Topics

- Authentication (p. 1388)
- Access control (p. 1389)
- Overview of managing access permissions to your AWS Control Tower resources (p. 1389)
- Prevent cross-service impersonation (p. 1396)
- Using identity-based policies (IAM policies) for AWS Control Tower (p. 1397)

Authentication

You have access to AWS as any of the following types of identities:

- **AWS account root user** – When you first create an AWS account, you begin with an identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user. You have access to this identity when you sign in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM Identity Center user (recommended) or IAM user (not a best practice in most use cases). Then securely lock away the root user credentials and use them to perform only a few account and service management tasks. For more information, see When to sign in as a root user (p. 41).

- **IAM user** – An IAM user is an identity within your AWS account that has specific, customized permissions. You can use the IAM user credentials to sign in to secure AWS webpages such as the AWS Management Console, AWS Discussion Forums, or the AWS Support Center. AWS best practices recommend that you create an IAM Identity Center user instead of an IAM user, because there is more security risk when you create an IAM user that has long-term credentials. If you must create an IAM user for a certain purpose, in addition to sign-in credentials, you can generate access keys for each IAM user. You can use these keys when you call AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don't use AWS tools, you must sign the request yourself. AWS Control Tower supports Signature Version 4, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the AWS General Reference.

- **IAM role** – An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity, and it has permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. IAM roles with temporary credentials are useful in the following situations:
  - **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as federated users. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated Users and Roles in the IAM User Guide.
  - **AWS service access** – A service role is an IAM role that a service assumes to perform actions in your account on your behalf. When you set up some AWS service environments, you must define a role for the service to assume. This service role must include all the permissions that are required for the service to access the AWS resources that it needs. Service roles vary from service to service, but many allow you to choose your permissions as long as you meet the documented requirements for that service. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. You can create, modify, and delete a service role from within IAM. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket.

If you must create an IAM user for a certain purpose, in addition to sign-in credentials, you can generate access keys for each IAM user. You can use these keys when you call AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don't use AWS tools, you must sign the request yourself. AWS Control Tower supports Signature Version 4, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the AWS General Reference.
on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see Creating a Role to Delegate Permissions to an AWS Service in the IAM User Guide.

- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an Amazon EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the Amazon EC2 instance. To assign an AWS role to an Amazon EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the Amazon EC2 instance to get temporary credentials. For more information, see Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances in the IAM User Guide.

- **IAM Identity Center user** Authentication to the IAM Identity Center user portal is controlled by the directory that you have connected to IAM Identity Center. However, authorization to the AWS accounts that are available to end users from within the user portal is determined by two factors:
  - Who has been assigned access to those AWS accounts in the AWS IAM Identity Center console. For more information, see Single Sign-On Access in the AWS IAM Identity Center User Guide.
  - What level of permissions have been granted to the end-users in the AWS IAM Identity Center console to allow them the appropriate access to those AWS accounts. For more information, see Permission Sets in the AWS IAM Identity Center User Guide.

### Access control

To create, update, delete, or list AWS Control Tower resources, or other AWS resources in your landing zone you need permissions to perform the operation, and you need permissions to access the corresponding resources. In addition, to perform the operation programmatically, you need valid access keys.

The following sections describe how to manage permissions for AWS Control Tower:

**Topics**

- Overview of managing access permissions to your AWS Control Tower resources (p. 1389)
- Using identity-based policies (IAM policies) for AWS Control Tower (p. 1397)

#### Overview of managing access permissions to your AWS Control Tower resources

Every AWS resource is owned by an AWS account, and permissions to create or gain access to a resource are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles). Some services (such as AWS Lambda) also support attaching permissions policies to resources.

**Note**

An account administrator (or administrator) is a user with administrator privileges. For more information, see IAM Best Practices in the IAM User Guide.

When you are responsible for granting permissions to a user or role, you must know and track the users and roles that require permissions, the resources for which each user and role requires permissions, and the specific actions that must be allowed for operating those resources.

**Topics**

- AWS Control Tower resources and operations (p. 1390)
- About resource ownership (p. 1390)
- Manage access to resources (p. 1390)
Overview of managing access with IAM

- Specifying policy elements: Actions, Effects, and Principals (p. 1396)
- Specifying conditions in a policy (p. 1396)

AWS Control Tower resources and operations

In AWS Control Tower, the primary resource is a landing zone. AWS Control Tower also supports an additional resource type, controls, sometimes referred to as guardrails. However, for AWS Control Tower, you can manage controls only in the context of an existing landing zone. Controls can be referred to as a subresource.

Resources and subresources in AWS have unique Amazon Resource Names (ARNs) associated with them, as shown in the following example.

About resource ownership

The AWS account owns the resources that are created in the account, regardless of who created the resources. Specifically, the resource owner is the AWS account of the principal entity (that is, the AWS account root user, an IAM Identity Center user, an IAM user, or an IAM role) that authenticates the resource creation request. The following examples illustrate how this works:

- If you use the AWS account root user credentials of your AWS account to set up a landing zone, your AWS account is the owner of the resource.
- If you create an IAM user in your AWS account and grant permissions to set up a landing zone to that user, the user can set up a landing zone as long as their account meets the prerequisites. However, your AWS account, to which the user belongs, owns the landing zone resource.
- If you create an IAM role in your AWS account with permissions to set up a landing zone, anyone who can assume the role can set up a landing zone. Your AWS account, to which the role belongs, owns the landing zone resource.

Manage access to resources

A permissions policy describes who has access to what. The following section explains the available options for creating permissions policies.

Note

This section discusses using IAM in the context of AWS Control Tower. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What Is IAM? in the IAM User Guide. For information about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the IAM User Guide.

Policies attached to an IAM identity are referred to as identity-based policies (IAM polices). Policies attached to a resource are referred to as resource-based policies.

Note

AWS Control Tower supports only identity-based policies (IAM policies).

Topics

- About identity-based policies (IAM policies) (p. 1390)
- Create roles and assign permissions (p. 1391)
- Resource-based policies (p. 1396)

About identity-based policies (IAM policies)

You can attach policies to IAM identities. For example, you can do the following:
• **Attach a permissions policy to a user or a group in your account** – To grant a user permissions to create an AWS Control Tower resource, such as setting up a landing zone, you can attach a permissions policy to a user or group that the user belongs to.

• **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, an administrator for one AWS account (Account A) can create a role that grants cross-account permissions to another AWS account (Account B), or the administrator can create a role that grants permissions to another AWS service.

  1. The Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions to manage resources in Account A.
  2. The Account A administrator attaches a trust policy to the role. The policy identifies Account B as the principal who can assume the role.
  3. As principal, the Account B administrator can give any user in Account B permission to assume the role. By assuming the role, users in Account B can create or gain access to resources in Account A.
  4. To grant an AWS service the ability (permissions) to assume the role, the principal that you specify in the trust policy can be an AWS service.

**Create roles and assign permissions**

Roles and permissions give you access to resources, in AWS Control Tower and in other AWS services, including programmatic access to resources.

To provide access, add permissions to your users, groups, or roles:

• Users and groups in AWS IAM Identity Center:

  Create a permission set. Follow the instructions in [Create a permission set](#) in the [AWS IAM Identity Center User Guide](#).

• Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in [Creating a role for a third-party identity provider (federation)](#) in the [IAM User Guide](#).

• IAM users:

  - Create a role that your user can assume. Follow the instructions in [Creating a role for an IAM user](#) in the [IAM User Guide](#).
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in [Adding permissions to a user (console)](#) in the [IAM User Guide](#).

For more information about using IAM to delegate permissions, see [Access Management](#) in the [IAM User Guide](#).

**Note**

When setting up an AWS Control Tower landing zone, you'll need a user or role with the **AdministratorAccess** managed policy. (arn:aws:iam::aws:policy/AdministratorAccess)

**To create a role for an AWS service (IAM console)**

1. Sign in to the AWS Management Console and open the IAM console at [https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. In the navigation pane of the IAM console, choose **Roles**, and then choose **Create role**.
3. Choose the **AWS service** role type.
4. Choose the use case for your service. Use cases are defined by the service to include the trust policy that the service requires.
5. Choose **Next**.
6. If possible, select the policy to use for the permissions policy. Otherwise, choose Create policy to open a new browser tab and create a new policy from scratch. For more information, see Creating IAM policies in the IAM User Guide.

7. After you create the policy, close that tab and return to your original tab. Select the check box next to the permissions policies that you want the service to have.

Depending on the use case that you selected, the service might let you do any of the following:

- Nothing, because the service defines the permissions for the role.
- Choose from a limited set of permissions.
- Choose from any permissions.
- Select no policies at this time. However, you can create the policies later, and then attach them to the role.

8. (Optional) Set a permissions boundary. This is an advanced feature that is available for service roles, but not for service-linked roles.

Expand the Permissions boundary section and choose Use a permissions boundary to control the maximum role permissions. IAM includes a list of the AWS managed and customer managed policies in your account. Select the policy to use for the permissions boundary or choose Create policy to open a new browser tab and create a new policy from scratch. For more information, see Creating IAM policies in the IAM User Guide. After you create the policy, close that tab and return to your original tab to select the policy to use for the permissions boundary.

9. Choose Next.

10. For Role name, the degree of role name customization is defined by the service. If the service defines the role's name, you can't edit this option. In other cases, the service might define a prefix for the role and you can enter an optional suffix. For some services, you can specify the entire name of your role.

If possible, enter a role name or role name suffix to help you identify the purpose of this role. Role names must be unique within your AWS account, so don't create roles named both PRODROLE and prodrole. When a role name is used in a policy or as part of an ARN, the role name is case sensitive. When a role name appears to customers in the console, such as during the sign-in process, the role name is case insensitive. Because various entities might reference the role, you can't edit the name of the role after it is created.

11. (Optional) For Description, enter a description for the new role.

12. Choose Edit in the Step 1: Select trusted entities or Step 2: Select permissions sections to edit the use cases and permissions for the role.

13. (Optional) Add metadata to the role by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM resources in the IAM User Guide.

14. Review the role, and then choose Create role.

To use the JSON policy editor to create a policy

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.

2. In the navigation pane on the left, choose Policies.

   If this is your first time choosing Policies, the Welcome to Managed Policies page appears. Choose Get Started.

3. At the top of the page, choose Create policy.

4. In the Policy editor section, choose the JSON option.

5. Enter or paste a JSON policy document. For details about the IAM policy language, see IAM JSON policy reference.
6. Resolve any security warnings, errors, or general warnings generated during policy validation, and then choose Next.

   **Note**
   You can switch between the Visual and JSON editor options anytime. However, if you make changes or choose Next in the Visual editor, IAM might restructure your policy to optimize it for the visual editor. For more information, see Policy restructuring in the IAM User Guide.

7. (Optional) When you create or edit a policy in the AWS Management Console, you can generate a JSON or YAML policy template that you can use in AWS CloudFormation templates.

   To do this, in the Policy editor choose Actions, and then choose Generate CloudFormation template. To learn more about AWS CloudFormation, see AWS Identity and Access Management resource type reference in the AWS CloudFormation User Guide.

8. When you are finished adding permissions to the policy, choose Next.

9. On the Review and create page, enter a Policy name and a Description (optional) for the policy that you are creating. Review Permissions defined in this policy to see the permissions that are granted by your policy.

10. (Optional) Add metadata to the policy by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM resources in the IAM User Guide.

11. Choose Create policy to save your new policy.

### To use the visual editor to create a policy

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane on the left, choose Policies.

   If this is your first time choosing Policies, the Welcome to Managed Policies page appears. Choose Get Started.

3. Choose Create policy.

4. In the Policy editor section, find the Select a service section, and then choose an AWS service. You can use the search box at the top to limit the results in the list of services. You can choose only one service within a visual editor permission block. To grant access to more than one service, add multiple permission blocks by choosing Add more permissions.

5. In Actions allowed, choose the actions to add to the policy. You can choose actions in the following ways:

   - Select the check box for all actions.
   - Choose Add actions to enter the name of a specific action. You can use a wildcard character (*) to specify multiple actions.
   - Select one of the Access level groups to choose all actions for the access level (for example, Read, Write, or List).
   - Expand each of the Access level groups to choose individual actions.

   By default, the policy that you are creating allows the actions that you choose. To deny the chosen actions instead, choose Switch to deny permissions. Because IAM denies by default, we recommend as a security best practice that you allow permissions to only those actions and resources that a user needs. Create a JSON statement to deny permissions only if you want to override a permission separately allowed by another statement or policy. We recommend that you limit the number of deny permissions to a minimum because they can increase the difficulty of troubleshooting permissions.

6. For Resources, if the service and actions that you selected in the previous steps do not support choosing specific resources, all resources are allowed and you cannot edit this section.
If you chose one or more actions that support resource-level permissions, then the visual editor lists those resources. You can then expand Resources to specify resources for your policy.

You can specify resources in the following ways:

- Choose Add ARNs to specify resources by their Amazon Resource Names (ARN). You can use the visual ARN editor or list ARNs manually. For more information about ARN syntax, see Amazon Resource Names (ARNs) in the IAM User Guide. For information about using ARNs in the Resource element of a policy, see IAM JSON policy elements: Resource in the IAM User Guide.
- Choose Any in this account next to a resource to grant permissions to any resources of that type.
- Choose All to choose all resources for the service.

7. (Optional) Choose Request conditions - optional to add conditions to the policy that you are creating. Conditions limit a JSON policy statement's effect. For example, you can specify that a user is allowed to perform the actions on the resources only when that user's request happens within a certain time range. You can also use commonly used conditions to limit whether a user must be authenticated by using a multi-factor authentication (MFA) device. Or you can require that the request originate from within a certain range of IP addresses. For lists of all of the context keys that you can use in a policy condition, see Actions, resources, and condition keys for AWS services in the Service Authorization Reference.

You can choose conditions in the following ways:

- Use check boxes to select commonly used conditions.
- Choose Add another condition to specify other conditions. Choose the condition's Condition Key, Qualifier, and Operator, and then enter a Value. To add more than one value, choose Add. You can consider the values as being connected by a logical OR operator. When you are finished, choose Add condition.

To add more than one condition, choose Add another condition again. Repeat as needed. Each condition applies only to this one visual editor permission block. All the conditions must be true for the permission block to be considered a match. In other words, consider the conditions to be connected by a logical AND operator.

For more information about the Condition element, see IAM JSON policy elements: Condition in the IAM User Guide.

8. To add more permission blocks, choose Add more permissions. For each block, repeat steps 2 through 5.

Note
You can switch between the Visual and JSON editor options anytime. However, if you make changes or choose Next in the Visual editor, IAM might restructure your policy to optimize it for the visual editor. For more information, see Policy restructuring in the IAM User Guide.

9. (Optional) When you create or edit a policy in the AWS Management Console, you can generate a JSON or YAML policy template that you can use in AWS CloudFormation templates.

To do this, in the Policy editor choose Actions, and then choose Generate CloudFormation template. To learn more about AWS CloudFormation, see AWS Identity and Access Management resource type reference in the AWS CloudFormation User Guide.

10. When you are finished adding permissions to the policy, choose Next.

11. On the Review and create page, enter a Policy name and a Description (optional) for the policy that you are creating. Review the Permissions defined in this policy to make sure that you have granted the intended permissions.

12. (Optional) Add metadata to the policy by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM resources in the IAM User Guide.
13. Choose **Create policy** to save your new policy.

**To grant programmatic access**

Users need programmatic access if they want to interact with AWS outside of the AWS Management Console. The way to grant programmatic access depends on the type of user that's accessing AWS.

To grant users programmatic access, choose one of the following options.

<table>
<thead>
<tr>
<th>Which user needs programmatic access?</th>
<th>To</th>
<th>By</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workforce identity (Users managed in IAM Identity Center)</td>
<td>Use temporary credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.</td>
<td>Following the instructions for the interface that you want to use.</td>
</tr>
<tr>
<td><strong>• For the AWS CLI,</strong> see <a href="https://docs.aws.amazon.com/cli/latest/userguide/cli-iam.html">Configuring the AWS CLI to use AWS IAM Identity Center</a> in the AWS Command Line Interface User Guide.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>• For AWS SDKs, tools, and AWS APIs,</strong> see <a href="https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/ssl-identity-center.html">IAM Identity Center authentication</a> in the AWS SDKs and Tools Reference Guide.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAM</td>
<td>Use temporary credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.</td>
<td>Following the instructions in <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/idConditions.html">Using temporary credentials with AWS resources</a> in the IAM User Guide.</td>
</tr>
<tr>
<td>IAM (Not recommended)</td>
<td>Use long-term credentials to sign programmatic requests to the AWS CLI, AWS SDKs, or AWS APIs.</td>
<td>Following the instructions for the interface that you want to use.</td>
</tr>
<tr>
<td><strong>• For the AWS CLI,</strong> see <a href="https://docs.aws.amazon.com/cli/latest/userguide/cli-iam-user-credentials.html">Authenticating using IAM user credentials</a> in the AWS Command Line Interface User Guide.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>• For AWS APIs,</strong> see <a href="https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html">Managing access keys for IAM users</a> in the IAM User Guide.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Protect against attackers**

For more information about how to help protect against attackers when you grant permissions to other AWS service principals, see [Optional conditions for your role trust relationships](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_permissions-policy-elements.html#optional-conditions) (p. 88). By adding certain conditions to your policies, you can help prevent a specific type of attack, known as a confused deputy attack, which occurs if an entity coerces a more-privileged entity to perform an action, such as
with cross-service impersonation. For general information about policy conditions, also see *Specifying conditions in a policy* (p. 1396).

For more information about using identity-based policies with AWS Control Tower, see *Using identity-based policies (IAM policies) for AWS Control Tower* (p. 1397). For more information about users, groups, roles, and permissions, see *Identities (Users, Groups, and Roles)* in the IAM User Guide.

**Resource-based policies**

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. AWS Control Tower does not support resource-based policies.

**Specifying policy elements: Actions, Effects, and Principals**

Currently, AWS Control Tower doesn't have an API for setting up a landing zone, only for managing controls. You can set up and manage your landing zone through the AWS Control Tower console. To set up your landing zone, you must be an IAM user with administrative permissions as defined in an IAM policy.

The following elements are the most basic ones you can identify in a policy:

- **Resource** – In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For more information, see *AWS Control Tower resources and operations* (p. 1390).
- **Action** – You use action keywords to identify resource operations that you want to allow or deny. For information about types of actions available to be performed, see *Actions defined by AWS Control Tower*.
- **Effect** – You specify the effect when the user requests the specific action—this can be either allow or deny. If you don't explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.
- **Principal** – In identity-based policies (IAM policies), that user to which the policy is attached is the implicit principal. For resource-based policies, you specify the user, account, service, or other entity that you want to receive permissions (applies to resource-based policies only). AWS Control Tower doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see *AWS IAM Policy Reference* in the IAM User Guide.

**Specifying conditions in a policy**

When you grant permissions, you can use the IAM policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see *Condition* in the IAM User Guide.

To express conditions, you can use predefined condition keys. There are no condition keys specific to AWS Control Tower. However, there are AWS-wide condition keys that you can use as appropriate. For a complete list of AWS-wide keys, see *Available Keys for Conditions* in the IAM User Guide.

**Prevent cross-service impersonation**

In AWS, cross-service impersonation can result in the confused deputy problem. When one service calls another service, cross-service impersonation occurs if one service manipulates another service to use its permissions to act on a customer's resources in a way that's not otherwise permitted. To prevent this
attack, AWS provides tools to help you protect your data, so that only those services with legitimate permission can gain access to resources in your account.

We recommend using the `aws:SourceArn` and `aws:SourceAccount` conditions in your policies, to limit the permissions that AWS Control Tower gives to another service for access to your resources.

- Use `aws:SourceArn` if you want only one resource to be associated with cross-service access.
- Use `aws:SourceAccount` if you want to allow any resource in that account to be associated with cross-service use.
- If the `aws:SourceArn` value does not contain the account ID, such as the ARN for an Amazon S3 bucket, you must use both conditions to limit permissions.
- If you use both conditions, and if the `aws:SourceArn` value contains the account ID, the `aws:SourceAccount` value and the account in the `aws:SourceArn` value must show the same account ID when used in the same policy statement.

For more information and examples, see Optional conditions for your role trust relationships (p. 88).

**Using identity-based policies (IAM policies) for AWS Control Tower**

This topic provides examples of identity-based policies that demonstrate how an account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles) and thereby grant permissions to perform operations on AWS Control Tower resources.

**Important**

We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your AWS Control Tower resources. For more information, see Overview of managing access permissions to your AWS Control Tower resources (p. 1389).

**Permissions Required to Use the AWS Control Tower Console**

AWS Control Tower creates three roles automatically when you set up a landing zone. All three roles are required to allow console access. AWS Control Tower splits permissions into three roles as a best practice to restrict access to the minimal sets of actions and resources.

**Three required roles**

- [AWSControlTowerAdmin role](p. 1397)
- [AWSControlTowerStackSetRole](p. 1401)
- [AWSControlTowerCloudTrailRole](p. 1402)

We recommend that you restrict access to your role trust policies for these roles. For more information, see Optional conditions for your role trust relationships (p. 88).

**AWSControlTowerAdmin role**

This role provides AWS Control Tower with access to infrastructure critical to maintaining the landing zone. The AWSControlTowerAdmin role requires an attached managed policy and a role trust policy for the IAM role. A role trust policy is a resource-based policy, specifying which principals can assume the role.

**Managed Policy for this role: AWSControlTowerServiceRolePolicy**
AWSControlTowerServiceRolePolicy

The AWSControlTowerServiceRolePolicy AWS-managed policy defines permissions to create and manage AWS Control Tower resources such as AWS CloudFormation stacksets and stack instances, AWS CloudTrail log files, a configuration aggregator for AWS Control Tower, as well as AWS Organizations accounts and organizational units (OUs) that are governed by AWS Control Tower.

Updates to this managed policy are summarized in the table, Managed policies for AWS Control Tower (p. 1405).

For more information, see AWSControlTowerServiceRolePolicy in the AWS Managed Policy Reference Guide.

Managed Policy Name: AWSControlTowerServiceRolePolicy

The JSON artifact for AWSControlTowerServiceRolePolicy is the following:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:CreateStackInstances",
        "cloudformation:CreateStackSet",
        "cloudformation:DeleteStack",
        "cloudformation:DeleteStackInstances",
        "cloudformation:DeleteStackSet",
        "cloudformation:DescribeStackInstance",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackSet",
        "cloudformation:DescribeStackSetOperation",
        "cloudformation:ListStackInstances",
        "cloudformation:UpdateStack",
        "cloudformation:UpdateStackInstances",
        "cloudformation:UpdateStackSet"
      ],
      "Resource": [
        "arn:aws:cloudformation:*::*:type/resource/AWS-IAM-Role"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "account:EnableRegion",
        "account:ListRegions",
        "account:GetRegionOptStatus"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:CreateStackInstances",
        "cloudformation:CreateStackSet",
        "cloudformation:DeleteStack",
        "cloudformation:DeleteStackInstances",
        "cloudformation:DeleteStackSet",
        "cloudformation:DescribeStackInstance",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackSet"
      ],
      "Resource": "*"
    }
  ]
}
```
Using identity-based policies (IAM policies)

```

"cloudformation:DescribeStackSetOperation",
"cloudformation:GetTemplate",
"cloudformation:ListStackInstances",
"cloudformation:UpdateStack",
"cloudformation:UpdateStackInstances",
"cloudformation:UpdateStackSet"
],
"Resource": [
  "arn:aws:cloudformation:*:*:stack/AWSControlTower*/*",
  "arn:aws:cloudformation:*:*:stack/StackSet-AWSControlTower*/*",
  "arn:aws:cloudformation:*:*:stackset/AWSControlTower*:*",
  "arn:aws:cloudformation:*:*:stackset-target/AWSControlTower*/*"
],
"Effect": "Allow",
"Action": [
  "cloudtrail:CreateTrail",
  "cloudtrail:DeleteTrail",
  "cloudtrail:GetTrailStatus",
  "cloudtrail:StartLogging",
  "cloudtrail:StopLogging",
  "cloudtrail:UpdateTrail",
  "cloudtrail:PutEventSelectors",
  "logs:CreateLogStream",
  "logs:PutLogEvents",
  "logs:PutRetentionPolicy"
],
"Resource": [
  "arn:aws:cloudtrail:*:*:trail/aws-controltower*
]
],
"Effect": "Allow",
"Action": [
  "sts:AssumeRole"
],
"Resource": [
  "arn:aws:iam::*:role/AWSControlTowerExecution",
  "arn:aws:iam::*:role/AWSControlTowerBlueprintAccess"
]
},
"Effect": "Allow",
"Action": [
  "cloudtrail:DescribeTrails",
  "ec2:DescribeAvailabilityZones",
  "iam:ListRoles",
  "logs:CreateLogGroup",
  "logs:DescribeLogGroups",
  "organizations:CreateAccount",
  "organizations:DescribeAccount",
  "organizations:DescribeCreateAccountStatus",
  "organizations:DescribeOrganization",
  "organizations:DescribeOrganizationalUnit",
  "organizations:DescribePolicy",
  "s3:GetObject",
  "sts:GetC current_page
Using identity-based policies (IAM policies)

```
"organizations:ListAccounts",
"organizations:ListAccountsForParent",
"organizations:ListAWSServiceAccessForOrganization",
"organizations:ListChildren",
"organizations:ListOrganizationalUnitsForParent",
"organizations:ListParents",
"organizations:ListPoliciesForTarget",
"organizations:ListTargetsForPolicy",
"organizations:ListRoots",
"organizations:MoveAccount",
"servicecatalog:AssociatePrincipalWithPortfolio"
],
"Resource": "*"
},
{
"Effect": "Allow",
"Action": [
  "iam:GetRole",
  "iam:GetUser",
  "iam:ListAttachedRolePolicies",
  "iam:GetRolePolicy"
],
"Resource": "*"
},
{
"Effect": "Allow",
"Action": [
  "iam:PassRole"
],
"Resource": [
  "arn:aws:iam::*:role/service-role/AWSControlTowerStackSetRole",
  "arn:aws:iam::*:role/service-role/AWSControlTowerCloudTrailRole",
  "arn:aws:iam::*:role/service-role/AWSControlTowerConfigAggregatorRoleForOrganizations"
]
},
{
"Effect": "Allow",
"Action": [
  "config:DeleteConfigurationAggregator",
  "config:PutConfigurationAggregator",
  "config:TagResource"
],
"Resource": "*",
"Condition": {
  "StringEquals": {
    "aws:ResourceTag/aws-control-tower": "managed-by-control-tower"
  }
}
},
{
"Effect": "Allow",
"Action": [
  "organizations:EnableAWSServiceAccess",
  "organizations:DisableAWSServiceAccess"
],
"Resource": "*",
"Condition": {
  "StringLike": {
    "organizations:ServicePrincipal": [
      "config.amazonaws.com",
      "cloudtrail.amazonaws.com"
    ]
  }
}
}]
```
Using identity-based policies (IAM policies)

Role trust policy:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": ["controltower.amazonaws.com"]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

The inline policy is `AWSControlTowerAdminPolicy`:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["sts:AssumeRole"],
      "Resource": ["arn:aws:iam::*:role/AWSControlTowerExecution"],
      "Effect": "Allow"
    }
  ]
}
```

**AWSControlTowerStackSetRole**

AWS CloudFormation assumes this role to deploy stack sets in accounts created by AWS Control Tower.

**Inline Policy:**

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["sts:AssumeRole"],
      "Resource": ["arn:aws:iam::*:role/AWSControlTowerExecution"],
      "Effect": "Allow"
    }
  ]
}
```
AWS Control Tower User Guide
Using identity-based policies (IAM policies)

AWSControlTowerCloudTrailRole

AWS Control Tower enables CloudTrail as a best practice and provides this role to CloudTrail. CloudTrail assumes this role to create and publish CloudTrail logs. Inline Policy:

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Action": "logs:CreateLogStream",
      "Effect": "Allow"
   },
   {
      "Action": "logs:PutLogEvents",
      "Effect": "Allow"
   }
   ]
}
```

AWSControlTowerBlueprintAccess role requirements

AWS Control Tower requires you to create the AWSControlTowerBlueprintAccess role in the designated blueprint hub account, within the same organization.

Role name

The role name must be AWSControlTowerBlueprintAccess.

Role trust policy

The role must be set up to trust the following principals:

- The principal that uses AWS Control Tower in the management account.
- The AWSControlTowerAdmin role in the management account.

The following example shows a least-privilege trust policy:

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Effect": "Allow",
      "Principal": {
         "AWS": [
           "arn:aws:iam::ManagementAccountId:role/AWSControlTowerAdmin",
           "arn:aws:iam::ManagementAccountId:role/YourControlTowerUserRole"
         ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
   }
   ]
}
```
Role permissions

You are required to attach the managed policy `AWSServiceCatalogAdminFullAccess` to the role.

**AWSServiceRoleForAWSControlTower**

This role provides AWS Control Tower with access to the Log Archive account, Audit account, and member accounts, for operations critical to maintaining the landing zone, such as notifying you of drifted resources.

The `AWSServiceRoleForAWSControlTower` role requires an attached managed policy and a role trust policy for the IAM role.

**Managed policy for this role:** `AWSControlTowerAccountServiceRolePolicy`

**Role trust policy:**

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "Service": "controltower.amazonaws.com"
         },
         "Action": "sts:AssumeRole"
      }
   ]
}
```

**AWSControlTowerAccountServiceRolePolicy**

This AWS-managed policy allows AWS Control Tower to call AWS services that provide automated account configuration and centralized governance on your behalf.

The policy contains the minimum permissions for AWS Control Tower to implement AWS Security Hub findings forwarding for resources managed by Security Hub controls that are part of the **Security Hub Service-managed Standard: AWS Control Tower**, and it prevents changes that restrict the ability to manage customer accounts. It is part of background AWS Security Hub drift detection process that is not directly initiated by a customer.

The policy gives permissions to create Amazon EventBridge rules, specifically for Security Hub controls, in each member account, and these rules must specify an exact EventPattern. Also, a rule can operate only on rules managed by our service principal.

**Service principal:** `controltower.amazonaws.com`

The JSON artifact for `AWSControlTowerAccountServiceRolePolicy` is the following:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowPutRuleOnSpecificSourcesAndDetailTypes",
         "Effect": "Allow",
         "Action": "events:PutRule",
         "Resource": "arn:aws:events:*::*:rule/*ControlTower*",
         "Condition": {
            "ForAnyValue:StringEquals": {
               "events:source": "aws.securityhub"
            }
         }
      }
   ]
}
```
Updates to this managed policy are summarized in the table, Managed policies for AWS Control Tower (p. 1405).
Managed policies for AWS Control Tower

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. Managed policies grant necessary permissions for common use cases so you can avoid having to investigate what permissions are needed. For more information, see AWS Managed Policies in the IAM User Guide.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWSControlTowerAccountServiceRolePolicy</td>
<td>AWS Control Tower added a new service-linked role that allows AWS Control Tower to create and manage event rules, and based on those rules, to manage drift detection for controls that are related to Security Hub. This change is needed so that customers can view drifted resources in the console, when those resources are related to Security Hub controls that are part of the Security Hub Service-managed Standard: AWS Control Tower.</td>
<td>May 22, 2023</td>
</tr>
<tr>
<td>AWSControlTowerServiceRolePolicy</td>
<td>AWS Control Tower added new permissions that allow AWS Control Tower to make calls to the EnableRegion, ListRegions, and GetRegionOptStatus APIs implemented by the AWS Account Management service, to make the opt-in AWS Regions available for customer accounts in the landing zone (Management account, Log archive account, Audit account, OU member accounts). This change is needed so that customers can have the option to expand Region governance by AWS Control Tower into the opt-in Regions.</td>
<td>April 6, 2023</td>
</tr>
<tr>
<td>AWSControlTowerServiceRolePolicy</td>
<td>AWS Control Tower added new permissions that allow AWS Control Tower to assume the AWSControlTowerBlueprintAccess role in the blueprint (hub) account, which is a dedicated account in an organization, containing pre-defined blueprints stored in one or more Service Catalog Products. AWS</td>
<td>October 28, 2022</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>Control Tower assumes the AWSControlTowerBlueprintAccess role to perform three tasks: create a Service Catalog Portfolio, add the requested blueprint Product, and share the Portfolio to a requested member account at account provisioning time. This change is needed so that customers can provision customized accounts through AWS Control Tower Account Factory.</td>
<td>June 20, 2022</td>
</tr>
<tr>
<td>AWSControlTowerServiceRolePolicy</td>
<td>AWS Control Tower added new permissions that allow customers to set up organization-level AWS CloudTrail trails, starting in landing zone version 3.0. The organization-based CloudTrail feature requires customers to have trusted access enabled for the CloudTrail service, and the IAM user or role must have permission to create an organization-level trail in the management account.</td>
<td>June 20, 2022</td>
</tr>
<tr>
<td>AWSControlTowerServiceRolePolicy</td>
<td>AWS Control Tower added new permissions that allow customers to use KMS key encryption. The KMS feature allows customers to provide their own KMS key to encrypt their CloudTrail logs. Customers also can change the KMS key during landing zone update or repair. When updating the KMS key, AWS CloudFormation needs permissions to call the AWS CloudTrail PutEventSelector API. The change to the policy is to allow the AWSControlTowerAdmin role to call the AWS CloudTrail PutEventSelector API.</td>
<td>July 28, 2021</td>
</tr>
<tr>
<td>AWS Control Tower started tracking changes</td>
<td>AWS Control Tower started tracking changes for its AWS managed policies.</td>
<td>May 27, 2021</td>
</tr>
</tbody>
</table>
Compliance Validation for AWS Control Tower

AWS Control Tower is a well-architected service that can help your organization meet your compliance needs with controls and best practices. Additionally, third-party auditors assess the security and compliance of a number of the services you can use in your landing zone as a part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact in the AWS Artifact User Guide.

Your compliance responsibility when using AWS Control Tower is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- **Security and Compliance Quick Start Guides** – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- **Architecting for HIPAA Security and Compliance on Amazon Web Services** – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- **AWS Compliance Resources** – This collection of workbooks and guides might apply to your industry and location.
- **AWS Config** – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- **AWS Security Hub** – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in AWS Control Tower

The AWS global infrastructure is built around AWS Regions and Availability Zones.

AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected by means of low-latency, high-throughput, and highly redundant networking. Availability Zones allow you to design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For a list of AWS Regions where AWS Control Tower is available, see How AWS Regions Work With AWS Control Tower (p. 97).

Your *home region* is defined as the AWS Region in which your landing zone was set up.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure Security in AWS Control Tower

AWS Control Tower is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls for access to AWS services and resources within your landing zone through the network. We require Transport Layer Security (TLS) 1.2 and recommend Transport Layer
Security (TLS) 1.3 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

You can set up security groups to provide additional network infrastructure security for your AWS Control Tower landing zone workloads. For more information, see Walkthrough: Set Up Security Groups in AWS Control Tower With AWS Firewall Manager (p. 1434).
Logging and monitoring in AWS Control Tower

Monitoring allows you to plan for and respond to potential incidents. Therefore, monitoring is an important part of the well-architected nature of AWS Control Tower. The results of monitoring activities are stored in log files; therefore, logging and monitoring are closely related concepts.

When you set up your landing zone, one of the shared accounts created is the log archive account, dedicated to collecting all logs centrally, including logs for all of your other accounts. These log files allow administrators and auditors to review actions and events that have occurred.

As a best practice, you should collect monitoring data from all of the parts of your AWS setup into your logs, so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your resources and activity in your landing zone.

For example, the status of your controls is monitored constantly. You can see their status at a glance in the AWS Control Tower console. The health and status of the accounts you provisioned in Account Factory also is monitored constantly.

Logging

AWS Control Tower accomplishes logging of actions and events automatically, through its integration with AWS CloudTrail and AWS Config, and it records them in CloudWatch. All actions are logged, including actions from the AWS Control Tower management account and from your organization's member accounts. Management account actions and events are viewable on the Activities page in the console. You can view member account actions and events in the log archive files.

The Activities Page

The Activities page provides an overview of AWS Control Tower management account actions. To navigate to the AWS Control Tower Activities page, select Activities from the left navigation.

The Activities page shows all AWS Control Tower actions initiated from the management account. It includes actions that are logged automatically when you navigate through the AWS Control Tower console. Here are the fields that the Activities page shows you:

- Date and time: The timestamp for the activity.
- User: The person or account that initiated the activity.
- Action: The activity that occurred.
- Resources: The resources affected by the activity.
- Status: Success, failure, or other state of the activity.
- Description: More details about the activity.

The activities shown in the Activities page are the same ones reported in the AWS CloudTrail events log for AWS Control Tower, but they're shown in a table format. To learn more about a specific activity, select the activity from the table and then choose View details.

The following sections describe monitoring and logging in AWS Control Tower with more detail:

Topics

- Monitoring (p. 1410)
Monitoring

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Control Tower and your other AWS solutions. AWS provides the following monitoring tools to watch AWS Control Tower, report when something is wrong, and take automatic actions when appropriate:

- **Amazon CloudWatch** monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the [Amazon CloudWatch User Guide](https://docs.aws.amazon.com/AmazonCloudWatch/latest/userguide/).

- **Amazon CloudWatch Events** delivers a near real-time stream of system events that describe changes in AWS resources. CloudWatch Events enables automated event-driven computing, as you can write rules that watch for certain events and trigger automated actions in other AWS services when these events happen. For more information, see the [Amazon CloudWatch Events User Guide](https://docs.aws.amazon.com/AmazonCloudWatchEvents/latest/userguide/).

- **Amazon CloudWatch Logs** enables you to monitor, store, and access your log files from Amazon EC2 instances, CloudTrail, and other sources. CloudWatch Logs can monitor information in the log files and notify you when certain thresholds are met. You can also archive your log data in highly durable storage. For more information, see the [Amazon CloudWatch Logs User Guide](https://docs.aws.amazon.com/AmazonCloudWatchLogs/latest/userguide/).

- **AWS CloudTrail** captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred.

Tip: You can view and query CloudTrail activity on an account through CloudWatch Logs and CloudWatch Logs Insights. This activity includes AWS Control Tower lifecycle events. CloudWatch Logs' capabilities allow you to perform more granular and precise queries than you would normally be able to make using CloudTrail.


Logging AWS Control Tower Actions with AWS CloudTrail

AWS Control Tower is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Control Tower. CloudTrail captures actions for AWS Control Tower as events. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Control Tower.

If you don't configure a trail, you can still view the most recent events in the CloudTrail console in Event history. Using the information collected by CloudTrail, you can determine the request that was made to AWS Control Tower, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, including how to configure and enable it, see the [AWS CloudTrail User Guide](https://docs.aws.amazon.com/CloudTrail/latest/userguide/).
AWS Control Tower Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in AWS Control Tower, that activity is recorded in a CloudTrail event along with other AWS service events in Event history. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

**Note**

In AWS Control Tower releases before landing zone version 3.0, AWS Control Tower created a member account trail. When you update to release 3.0, your CloudTrail trail is updated to become an organization trail. For best practices when moving between trails, see Creating an organizational trail in the CloudTrail User Guide.

**Recommended: Create a trail**

For an ongoing record of events in your AWS account, including events for AWS Control Tower, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- Prepare for creating a trail
- Managing CloudTrail costs
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

AWS Control Tower logs the following actions as events in CloudTrail log files:

**Public APIs**

- DisableControl
- EnableControl
- GetControlOperation
- ListEnabledControls

**Other APIs**

- SetupLandingZone
- UpdateAccountFactoryConfig
- ManageOrganizationalUnit
- CreateManagedAccount
- EnableGuardrail
- GetLandingZoneStatus
- GetHomeRegion
- ListManagedAccounts
- DescribeManagedAccount
- DescribeAccountFactoryConfig
Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.
- Whether the request was rejected as access denied or processed successfully.

For more information, see the CloudTrail `userIdentity` Element.

**Example: AWS Control Tower Log File Entries**

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail events don't appear in any specific order in the log files.

The following example shows a CloudTrail log entry that shows the structure of a typical log file entry for a `SetupLandingZone` AWS Control Tower event, including a record of the identity of the user who initiated the action.

```json
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:backend-test-assume-role-session",
    "arn": "arn:aws:sts::76543EXAMPLE::assumed-role/AWSControlTowerTestAdmin/backend-test-assume-role-session",
    "accountId": "76543EXAMPLE",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
```

1412
Monitoring resource changes with AWS Config

AWS Control Tower enables AWS Config on all enrolled accounts, so that it can monitor compliance through detective controls, record resource changes, and deliver resource change logs to the log archive account.

If your landing zone version is earlier than 3.0: For your enrolled accounts, AWS Config logs all changes to resources, for all Regions in which the account operates. Each change is modeled as a configuration item (CI), which contains information such as the resource identifier, the Region, the date that each change was recorded, and whether the change relates to a known resource or a newly discovered one.

If your landing zone version is 3.0 or later: AWS Control Tower limits recording for global resources, such as IAM users, groups, roles, and customer managed policies, to your home Region only. Copies of global resource changes are not stored in every Region. This limitation of resource recording conforms with AWS Config best practices. A full list of global resources is available in AWS Config documentation.

- To learn more about AWS Config, see How AWS Config works.
- For a list of resources that AWS Config can support, see Supported resource types.
- To learn about how to customize resource tracking in the AWS Control Tower environment, see the blog post entitled Customize AWS Config resource tracking in AWS Control Tower.

AWS Control Tower sets up an AWS Config delivery channel in all enrolled accounts. Through this delivery channel, it logs all changes recorded by AWS Config in the log archive account, where they are stored to a folder in an Amazon Simple Storage Service bucket.
Managing AWS Config costs in AWS Control Tower

This section describes how AWS Config records and bills you for changes to resources in your AWS Control Tower accounts. This information may help you understand how to manage the costs associated with AWS Config, when you're utilizing AWS Control Tower. AWS Control Tower adds no additional cost.

**Note**

If your landing zone version is 3.0 or later: AWS Control Tower limits AWS Config recording for global resources, such as IAM users, groups, roles, and customer-managed polices, to your home Region only. Therefore, some of the information in this section may not apply to your landing zone.

AWS Config is designed to record each change to each resource, in each Region where an account operates, as a configuration item (CI). AWS Config bills you for each configuration item that it generates.

**How AWS Config operates**

AWS Config records resources in each Region, separately. Some global resources, such as IAM roles, are recorded once per Region. For example, if you create a new IAM role in an enrolled account that is operating in five Regions, AWS Config generates five CIs, one for each Region. Other global resources, such as Route 53 hosted zones, are recorded only once across all Regions. For example, if you create a new Route 53 hosted zone in an enrolled account, AWS Config generates one CI, regardless of how many Regions are selected for that account. For a list that helps you distinguish these types of resources, see the AWS Config documentation.

**Note**

When AWS Control Tower works with AWS Config, a Region may be governed by AWS Control Tower, or ungoverned, and AWS Config still records the changes if the account operates in that Region.

**AWS Config detects two types of relationships in resources**

AWS Config makes a distinction between *direct* and *indirect* relationships among resources. If a resource is returned in another resource's Describe API call, those resources are recorded as a direct relationship. When you change a resource in a direct relationship with another resource, AWS Config does not make a CI for both resources.

For example, if you create an Amazon EC2 instance, and the API requires you to create a network interface, AWS Config considers the Amazon EC2 instance to have a direct relationship with the network interface. As a result, AWS Config generates only one CI.

AWS Config records separate changes for resource relationships that are *indirect* relationships. For example, AWS Config generates two CIs if you create a security group and add an associated Amazon EC2 instance that's part of the security group.

For more information about direct and indirect relationships, see [What is a direct and an indirect relationship with respect to a resource?](https://docs.aws.amazon.com/config/latest/developerguide/aws-config-relationships.html)

You can find a list of resource relationships in the AWS Config documentation.

**View the AWS Config recorder data on enrolled accounts**

AWS Config is integrated with CloudWatch so that you can view AWS Config CIs in a dashboard. For more information, see the blog post entitled [AWS Config supports Amazon CloudWatch metrics](https://aws.amazon.com/blogs/aws/aws-config-supports-amazon-cloudwatch-metrics/).

Programmatically, to view AWS Config data, you can work with the AWS CLI, or you can utilize other AWS tools.
Query the AWS Config recorder data on a specific resource

You can use the AWS CLI to retrieve a list of the most recent changes for a resource.

Resource history command:

- `aws configservice get-resource-config-history --resource-type RESOURCE-TYPE --resource-id RESOURCE-ID --region REGION`

To learn more, see the API documentation for `get-config-history`.

Visualize AWS Config data with Amazon QuickSight

You can visualize and query resources recorded by AWS Config across your entire organization. For more information, see Visualizing AWS Config data using Amazon Athena and Amazon QuickSight.

Troubleshooting AWS Config in AWS Control Tower

This section gives information about some problems you may encounter when using AWS Config with AWS Control Tower.

High AWS Config costs

If your workflow includes processes that create, update, or delete resources frequently, or if it handles resources in large numbers, that workflow may generate large numbers of CIs. If you run these processes in a non-production account, consider unenrolling the account. You may need to de-activate the AWS Config recorder for that account manually.

Note
After you unenroll the account, AWS Control Tower cannot enforce detective controls or log account events, such as AWS Config activities, for resources in that account.

For more information, see Unmanage an enrolled account. To learn how to deactivate the AWS Config recorder, see Managing the configuration recorder.

The same resource is recorded multiple times

Check whether the resource is a global resource. For AWS Control Tower landing zones prior to version 3.0, AWS Config may record certain global resources once for each Region in which AWS Config is operating. For example, if AWS Config is enabled on eight Regions, each role is recorded eight times.

The following resources are recorded once for each Region in which AWS Config is operating:

- AWS::IAM::Group
- AWS::IAM::Policy
- AWS::IAM::Role
- AWS::IAM::User

Other global resources are recorded only once. Here are some examples of resources that are recorded once:

- AWS::Route53::HostedZone
- AWS::Route53::HealthCheck
- AWS::ECR::PublicRepository
• AWS::GlobalAccelerator::Listener
• AWS::GlobalAccelerator::EndpointGroup
• AWS::GlobalAccelerator::Accelerator

AWS Config did not record a resource

Certain resources have dependency relationships with other resources. These relationships may be direct or indirect. You can find a list of deprecated indirect relationships in the AWS Config FAQ.

Lifecycle Events in AWS Control Tower

Some events logged by AWS Control Tower are lifecycle events. A lifecycle event's purpose is to mark the completion of certain AWS Control Tower actions that change the state of resources. Lifecycle events apply to resources that AWS Control Tower creates or manages, such as organizational units (OUs), accounts, and controls.

Characteristics of AWS Control Tower lifecycle events

• For each lifecycle event, the event log shows whether the originating Control Tower action completed successfully, or failed.
• AWS CloudTrail automatically records each lifecycle event as a non-API AWS service event. For more information, see the AWS CloudTrail User Guide.
• Each lifecycle event also is delivered to the Amazon EventBridge and Amazon CloudWatch Events services.

Lifecycle events in AWS Control Tower offer two primary benefits:

• Because a lifecycle event registers the completion of an AWS Control Tower action, you can create an Amazon EventBridge rule or Amazon CloudWatch Events rule that can trigger the next steps in your automation workflow, based on the state of the lifecycle event.
• The logs provide additional detail to assist administrators and auditors in reviewing certain types of activity in your organizations.

How lifecycle events work

AWS Control Tower relies upon multiple services to implement its actions. Therefore, each lifecycle event is recorded only after a series of actions is complete. For example, when you enable a control on an OU, AWS Control Tower launches a series of sub-steps that implement the request. The final result of the entire series of sub-steps is recorded in the log as the state of the lifecycle event.

• If every underlying sub-step has completed successfully, the lifecycle event state is recorded as Succeeded.
• If any of the underlying sub-steps did not complete successfully, the lifecycle event state is recorded as Failed.

Each lifecycle event includes a logged timestamp that shows when the AWS Control Tower action was initiated, and another timestamp showing when the lifecycle event is completed, marking success or failure.

Viewing lifecycle events in Control Tower

You can view lifecycle events from the Activities page in your AWS Control Tower dashboard.
To navigate to the **Activities** page, choose **Activities** from the left navigation pane.

To get more details about a specific event, select the event and then choose the **View details** button at the upper right.

For more information about how to integrate AWS Control Tower lifecycle events into your workflows, see this blog post, [*Using lifecycle events to track AWS Control Tower actions and trigger automated workflows*](#).

**Expected behavior of CreateManagedAccount and UpdateManagedAccount lifecycle events**

When you create an account or enroll an account in AWS Control Tower, those two actions call the same internal API. If there's an error during the process, it usually occurs after the account has been created but is not fully provisioned. When you retry to create the account after the error, or when you try to update the provisioned product, AWS Control Tower sees that the account already exists.

Because the account exists, AWS Control Tower records the **UpdateManagedAccount** lifecycle event instead of the **CreateManagedAccount** lifecycle event at the end of the retry request. You may have expected to see another **CreateManagedAccount** event because of the error. However, the **UpdateManagedAccount** lifecycle event is the expected and desired behavior.

If you plan to create or enroll accounts into AWS Control Tower using automated methods, program the Lambda function to look for **UpdateManagedAccount** lifecycle events as well as **CreateManagedAccount** lifecycle events.

**Lifecycle event names**

Each lifecycle event is named so that it corresponds to the originating AWS Control Tower action, which also is recorded by AWS CloudTrail. Thus, for example, a lifecycle event originated by the AWS Control Tower **CreateManagedAccount** CloudTrail event is named **CreateManagedAccount**.

Each name in the list that follows is a link to an example of the logged detail in JSON format. The additional detail shown in these examples is taken from the Amazon CloudWatch event logs.

Although JSON does not support comments, some comments have been added in the examples for explanatory purposes. Comments are preceded by “//” and they appear in the right side of the examples.

In these examples, some account names and organization names are obscured. An **accountId** is always a 12-number sequence, which has been replaced with “xxxxxxxxxxxx” in the examples. An **organizationalUnitID** is a unique string of letters and numbers. Its form is preserved in the examples.

- **CreateManagedAccount** *(p. 1418)*: The log records whether AWS Control Tower successfully completed every action to create and provision a new account using account factory.
- **UpdateManagedAccount** *(p. 1419)*: The log records whether AWS Control Tower successfully completed every action to update a provisioned product that's associated with an account you had previously created by using account factory.
- **EnableGuardrail** *(p. 1419)*: The log records whether AWS Control Tower successfully completed every action to enable a control on an OU that was created by AWS Control Tower.
- **DisableGuardrail** *(p. 1420)*: The log records whether AWS Control Tower successfully completed every action to disable a control on an OU that was created by AWS Control Tower.
- **SetupLandingZone** *(p. 1421)*: The log records whether AWS Control Tower successfully completed every action to set up a landing zone.
- **UpdateLandingZone** *(p. 1422)*: The log records whether AWS Control Tower successfully completed every action to update your existing landing zone.
- **RegisterOrganizationalUnit** *(p. 1424)*: The log records whether AWS Control Tower successfully completed every action to enable its governance features on an OU.
The log records whether AWS Control Tower successfully completed every action to disable its governance features on an OU.

- **DeregisterOrganizationalUnit** (p. 1424): The log records whether AWS Control Tower detected any resource that would prevent the **Extend governance** operation from completing successfully.

The following sections provide a list of AWS Control Tower lifecycle events, with examples of the details logged for each type of lifecycle event.

**CreateManagedAccount**

This lifecycle event records whether AWS Control Tower successfully created and provisioned a new account using account factory. This event corresponds to the AWS Control Tower CreateManagedAccount CloudTrail event. The lifecycle event log includes the accountName and accountId of the newly-created account, and the organizationalUnitName and organizationalUnitId of the OU in which the account has been placed.

```json
{
    "version": "0",
    "id": "999cccaaeaaa-0000-1111-123456789012",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "XXXXXXXXXXXX", // Management account ID.
    "region": "us-east-1", // AWS Control Tower home region.
    "resources": [],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "XXXXXXXXXXXX",
            "invokedBy": "AWS Internal"
        },
        "eventSource": "controltower.amazonaws.com",
        "eventName": "CreateManagedAccount",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "0000000-0000-0000-1111-123456789012",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceName": "controltower.amazonaws.com",
        "serviceEventDetails": {
            "createManagedAccountStatus": {
                "organizationalUnit": {
                    "organizationalUnitName": "Custom",
                    "organizationalUnitId": "ou-XXXX-l3zc8b3h"
                },
                "account": {
                    "accountName": "LifeCycle1",
                    "accountId": "XXXXXXXXXXXX"
                },
                "state": "SUCCEEDED",
                "message": "AWS Control Tower successfully created a managed account.",
                "requestedTimestamp": "2019-11-15T11:45:18+0000",
                "completedTimestamp": "2019-11-16T12:09:32+0000"
            }
        }
    }
}
```
UpdateManagedAccount

This lifecycle event records whether AWS Control Tower successfully updated the provisioned product associated with an account that was created previously by using account factory. This event corresponds to the AWS Control Tower UpdateManagedAccount CloudTrail event. The lifecycle event log includes the accountName and accountId of the associated account, and the organizationalUnitName and organizationalUnitId of the OU in which the updated account is placed.

```
{
  "version": "0",
  "id": "999ccaa-eaaa-0000-1111-123456789012",
  "detail-type": "AWS Service Event via CloudTrail",
  "source": "aws.controltower",
  "account": "XXXXXXXXXXXX",
  "region": "us-east-1", // AWS Control Tower home region.
  "resources": [],
  "detail": {
    "eventVersion": "1.05",
    "userIdentity": {
      "accountId": "XXXXXXXXX",
      "invokedBy": "AWS Internal"
    },
    "eventSource": "controltower.amazonaws.com",
    "eventName": "UpdateManagedAccount",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "eventID": "0000000-0000-0000-1111-123456789012",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "serviceEventDetails": {
      "updateManagedAccountStatus": {
        "organizationalUnit": {
          "organizationalUnitName": "Custom",
          "organizationalUnitId": "ou-XXXX-l3zc8b3h"
        },
        "account": {
          "accountName": "LifeCycle1",
          "accountId": "624281831893"
        },
        "state": "SUCCEEDED",
        "message": "AWS Control Tower successfully updated a managed account.",
        "requestedTimestamp": "2019-11-15T11:45:18+0000",
        "completedTimestamp": "2019-11-16T12:09:32+0000"
      }
    }
  }
}
```

EnableGuardrail

This lifecycle event records whether AWS Control Tower successfully enabled a control on an OU that is being managed by AWS Control Tower. This event corresponds to the AWS Control Tower EnableGuardrail CloudTrail event. The lifecycle event log includes the
DisableGuardrail

guardrailId and guardrailBehavior of the control, and the organizationalUnitName and organizationalUnitId of the OU on which the control is enabled.

```
{
  "version": "0",
  "id": "999ccaa-aaaa-0000-1111-123456789012",
  "detail-type": "AWS Service Event via CloudTrail",
  "source": "aws.controltower",
  "account": "XXXXXXXXXXXX",
  "region": "us-east-1", // AWS Control Tower home region.
  "resources": [],
  "detail": {
    "eventVersion": "1.05",
    "userIdentity": {
      "accountId": "XXXXXXXXXXXX",
      "invokedBy": "AWS Internal"
    },
    "eventTime": "2018-08-30T21:42:18Z",
    "eventSource": "controltower.amazonaws.com",
    "eventName": "EnableGuardrail",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "eventID": "0000000-0000-0000-1111-123456789012",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "serviceEventDetails": {
      "enableGuardrailStatus": {
        "organizationalUnits": [
          {
            "organizationalUnitName": "Custom",
            "organizationalUnitId": "ou-vwxy-18vy4yro"
          }
        ],
        "guardrails": [
          {
            "guardrailId": "AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK",
            "guardrailBehavior": "DETECTIVE"
          }
        ],
        "state": "SUCCEEDED",
        "message": "AWS Control Tower successfully enabled a guardrail on an organizational unit."
      }
    },
    "requestTimestamp": "2019-11-12T09:01:07+0000",
    "completedTimestamp": "2019-11-12T09:01:54+0000"
  }
}
```

DisableGuardrail

This lifecycle event records whether AWS Control Tower successfully disabled a control on an OU that is being managed by AWS Control Tower. This event corresponds to the AWS Control Tower DisableGuardrail CloudTrail event. The lifecycle event log includes the guardrailId and guardrailBehavior of the control, and the organizationalUnitName and organizationalUnitId of the OU on which the control is disabled.

```
{
  "version": "0",
  "id": "999ccaa-aaaa-0000-1111-123456789012",
  "detail-type": "AWS Service Event via CloudTrail",
  "source": "aws.controltower",
  "account": "XXXXXXXXXXXX",
  "time": "2018-08-30T21:42:18Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "eventVersion": "1.05",
    "userIdentity": {
      "accountId": "XXXXXXXXXXXX",
      "invokedBy": "AWS Internal"
    },
    "eventTime": "2018-08-30T21:42:18Z",
    "eventSource": "controltower.amazonaws.com",
    "eventName": "EnableGuardrail",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "eventID": "0000000-0000-0000-1111-123456789012",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "serviceEventDetails": {
      "enableGuardrailStatus": {
        "organizationalUnits": [
          {
            "organizationalUnitName": "Custom",
            "organizationalUnitId": "ou-vwxy-18vy4yro"
          }
        ],
        "guardrails": [
          {
            "guardrailId": "AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK",
            "guardrailBehavior": "DETECTIVE"
          }
        ],
        "state": "SUCCEEDED",
        "message": "AWS Control Tower successfully enabled a guardrail on an organizational unit."
      }
    },
    "requestTimestamp": "2019-11-12T09:01:07+0000",
    "completedTimestamp": "2019-11-12T09:01:54+0000"
  }
}
```

1420
SetupLandingZone

This lifecycle event records whether AWS Control Tower successfully set up a landing zone. This event corresponds to the AWS Control Tower SetupLandingZone CloudTrail event. The lifecycle event log includes the rootOrganizationalId, which is ID of the organization that AWS Control Tower creates from the management account. The log entry also includes the organizationalUnitName and organizationalUnitId for each of of the OUs, and the accountName and accountId for each account, that are created when AWS Control Tower sets up the landing zone.

```json
{
  "version": "0",
  "id": "999cccaaa-eaaa-0000-1111-123456789012",
  "detail-type": "AWS Service Event via CloudTrail",
  "source": "aws.controltower",
  "account": "XXXXXXXXXXXX",
  "time": "2018-08-30T21:42:18Z",
  "region": "us-east-1",
  "resources": [ ],
  "detail": {
    "eventVersion": "1.05",
    "userIdentity": {
      "accountId": "XXXXXXXXXXXX",
      "invokedBy": "AWS Internal"
    },
    "eventTime": "2018-08-30T21:42:18Z",
    "eventSource": "controltower.amazonaws.com",
    "eventName": "DisableGuardrail",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "eventID": "0000000-0000-0000-1111-123456789012",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "serviceName": "Control Tower",
    "serviceEventDetails": {
      "disableGuardrailId": "AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK",
      "disableGuardrailBehavior": "DETECTIVE",
      "organizationalUnits": [
        {
          "organizationalUnitName": "Custom",
          "organizationalUnitId": "ou-wwxy-xyzzy-zz",
          "state": "SUCCEEDED",
          "message": "AWS Control Tower successfully disabled a guardrail on an organizational unit.",
          "requestTimestamp": "2019-11-12T09:01:07+0000",
          "completedTimestamp": "2019-11-12T09:01:54+0000"
        }
      ]
    }
  }
}
```
UpdateLandingZone

This lifecycle event records whether AWS Control Tower successfully updated your existing landing zone. This event corresponds to the AWS Control Tower UpdateLandingZone CloudTrail event. The lifecycle event log includes the rootOrganizationalId, which is ID of the (updated) organization.
governed by AWS Control Tower. The log entry also includes the organizationalUnitName and organizationalUnitId for each of the OUs, and the accountName and accountId for each account, that was created previously, when AWS Control Tower originally set up the landing zone.

```json
{
    "version": "0",
    "id": "099cccaa-eaaa-0000-1111-123456789012", // Request ID.
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "XXXXXXXXXXXX", // Management account ID.
    "region": "us-east-1", // Management account CloudTrail region.
    "resources": [],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "XXXXXXXXXXXX", // Management account ID.
            "invokedBy": "AWS Internal"
        },
        "eventSource": "controltower.amazonaws.com",
        "eventName": "UpdateLandingZone",
        "awsRegion": "us-east-1", // AWS Control Tower home region.
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "CloudTrail_event_ID", // This value is generated by CloudTrail.
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "updateLandingZoneStatus": {
                "state": "SUCCEEDED", // Status of entire operation.
                "message": "AWS Control Tower successfully updated a landing zone."
            }
        }
    }
}
```

"rootOrganizationalId" : "r-1234",
"organizationalUnits": [ // Use a list.
    {
        "organizationalUnitName": "Security", // Security OU name.
        "organizationalUnitId": "ou-adpf-302pk332" // Security OU ID.
    },
    {
        "organizationalUnitName": "Custom", // Custom OU name.
        "organizationalUnitId": "ou-adpf-302pk332" // Custom OU ID.
    }
],
"accounts": [ // All created accounts are here. Use a list of "account" objects.
    {
        "accountName": "Audit",
        "accountId": "XXXXXXXXXXXX"
    },
    {
        "accountName": "Log archive",
        "accountId": "XXXXXXXXXXXX"
    }
],
"requestedTimestamp": "2018-08-30T21:42:18Z",

"1423"
RegisterOrganizationalUnit

This lifecycle event records whether AWS Control Tower successfully enabled its governance features on an OU. This event corresponds to the AWS Control Tower RegisterOrganizationalUnit CloudTrail event. The lifecycle event log includes the organizationalUnitName and organizationalUnitId of the OU that AWS Control Tower has brought under its governance.

DeregisterOrganizationalUnit

This lifecycle event records whether AWS Control Tower successfully disabled its governance features on an OU. This event corresponds to the AWS Control Tower DeregisterOrganizationalUnit event.
CloudTrail event. The lifecycle event log includes the organizationalUnitName and organizationalUnitId of the OU on which AWS Control Tower has disabled its governance features.

```json
{
    "version": "0",
    "id": "999ccaa-eaaa-0000-1111-123456789012",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "XXXXXXXXXXXX",
    "time": "2018-08-30T21:42:18Z",
    "region": "us-east-1",
    "resources": [ ],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "XXXXXXXXXXXX",
            "invokedBy": "AWS Internal"
        },
        "eventTime": "2018-08-30T21:42:18Z",
        "eventSource": "controltower.amazonaws.com",
        "eventName": "DeregisterOrganizationalUnit",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "0000000-0000-0000-1111-123456789012",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "deregisterOrganizationalUnitStatus": {
                "state": "SUCCEEDED",
                "message": "AWS Control Tower successfully deregistered an organizational unit, and enabled mandatory guardrails on the new organizational unit.",
                "organizationalUnit": {
                    "organizationalUnitName": "Test", // Foundational OU
                    "organizationalUnitId": "ou-adpf-302pk332" // Foundational OU
                },
                "requestedTimestamp": "2018-08-30T21:42:18Z",
                "completedTimestamp": "2018-08-30T21:42:18Z"
            }
        }
    }
}
```

**PrecheckOrganizationalUnit**

This lifecycle event records whether AWS Control Tower successfully performed prechecks on an OU. This event corresponds to the AWS Control Tower PrecheckOrganizationalUnit CloudTrail event. The lifecycle event log contains a field for the Id, Name, and failedPrechecks values, for each resource on which AWS Control Tower has performed prechecks during the OU registration process.

The event log also contains information about the nested accounts on which the prechecks were performed, including the accountName, accountId, and failedPrechecks fields.

If the failedPrechecks value is empty, it means that all prechecks for that resource passed successfully.

- This event is emitted only if there is a precheck failure.
- This event is not emitted if you are registering an empty OU.
Example of event:

```json
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "XXXXXXXXXXXX",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-09-20T22:45:43Z",
    "eventSource": "controltower.amazonaws.com",
    "eventName": "PrecheckOrganizationalUnit",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "eventID": "b41a9d67-0da4-4dc5-a87a-25fa19dc5305",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "XXXXXXXXXXXX",
    "serviceEventDetails": {
        "precheckOrganizationalUnitStatus": {
            "organizationalUnit": {
                "organizationalUnitName": "Ou-123",
                "organizationalUnitId": "ou-abcd-123456",
                "failedPrechecks": [
                    "SCP_CONFLICT"
                ]
            },
            "accounts": [
                { "accountName": "Child Account 1",
                  "accountId": "XXXXXXXXXXXX",
                  "failedPrechecks": [
                    "FAILED_TO_ASSUME_ROLE"
                ],
                { "accountName": "Child Account 2",
                  "accountId": "XXXXXXXXXXXX",
                  "failedPrechecks": [
                    "FAILED_TO_ASSUME_ROLE"
                ],
                { "accountName": "Management Account",
                  "accountId": "XXXXXXXXXXXX",
                  "failedPrechecks": [
                    "MISSING_PERMISSIONS_AF_PRODUCT"
                  ],
                { "accountName": "Child Account 3",
                  "accountId": "XXXXXXXXXXXX",
                  "failedPrechecks": []
                },
                ...
            ],
            "state": "FAILED",
            "message": "AWS Control Tower failed to register an organizational unit due to precheck failures. Go to the OU details page to download a list of failed pre-checks for the OU and accounts within."
        },
        "requestedTimestamp": "2021-09-20T22:44:02+0000",
        "completedTimestamp": "2021-09-20T22:45:43+0000"
    }
}
```
Using AWS User Notifications with AWS Control Tower

You can use AWS User Notifications to set up delivery channels to be notified about AWS Control Tower events. You receive a notification when an event matches a rule that you specify. You can receive notifications for events through multiple channels, including email, AWS Chatbot chat notifications, or AWS Console Mobile App push notifications. You can also see notifications in the Console Notifications Center.

AWS User Notifications supports aggregation, which can reduce the number of notifications you receive during specific events. Notifications also are visible in the Console Notifications Center.

The advantages of subscribing to notifications through AWS User Notifications instead of EventBridge include:

- A friendlier user interface (UI).
- Integration with the AWS console, in the bell/notifications area on the global navigation bar.
- Native support for email notifications, there's no need to set up Amazon SNS.
- Most notably, support for mobile push notifications, exclusive to AWS User Notifications.

For example, one type of notification you may wish to receive is in case of Security Hub critical and high severity findings. A code snippet in JSON to set up that notification subscription may look something like this:

```json
{
  "detail": {
    "findings": {
      "Compliance": {
        "Status": ["FAILED", "WARNING", "NOT_AVAILABLE"]
      },
      "RecordState": ["ACTIVE"],
      "Severity": {
        "Label": ["CRITICAL", "HIGH"]
      },
      "Workflow": {
        "Status": ["NEW", "NOTIFIED"]
      }
    }
  }
}
```

**Event filtering**

- You can filter events by service and name using the filters available on the AWS User Notifications console.
- You can filter events by specific properties if you create your own EventBridge filter from JSON code.

**Example AWS Control Tower event**

Here is a generalized example event for AWS Control Tower.
• It an EventBridge event.
• You can subscribe to EventBridge events (such as this one) using AWS User Notifications.

```json
{
  "version": "0",
  "id": "<id>", // alphanumeric string
  "detail-type": "AWS Service Event via CloudTrail",
  "source": "aws.controltower",
  "account": "<account ID>", // Management account ID.
  "time": "<date>", // Format: yyyy-MM-dd'T'hh:mm:ssZ
  "region": "<region>", // AWS Control Tower home region.
  "resources": [],
  "detail": {
    "eventVersion": "1.05",
    "userIdentity": {
      "accountId": "121212121212",
      "invokedBy": "AWS Internal"
    },
    "eventSource": "controltower.amazonaws.com",
    "eventName": "<event name>", // one of the 9 event names in https://docs.aws.amazon.com/controltower/latest/userguide/lifecycle-events.html
    "awsRegion": "<region>",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "eventID": "<id>",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "serviceEventDetails": {
      // the contents of this object vary depending on the event subtype and event state
    }
  }
}
```
Walkthroughs

This chapter contains walkthrough procedures that can help you in your use of AWS Control Tower.

Topics

- Walkthrough: Move from ALZ to AWS Control Tower (p. 1429)
- Walkthrough: Automate Account Provisioning in AWS Control Tower by Service Catalog APIs (p. 1429)
- Walkthrough: Configure AWS Control Tower Without a VPC (p. 1432)
- Manage AWS Control Tower Resources (p. 1437)
- Walkthrough: Set Up Security Groups in AWS Control Tower With AWS Firewall Manager (p. 1434)
- Walkthrough: Decommission an AWS Control Tower Landing Zone (p. 1434)

Walkthrough: Move from ALZ to AWS Control Tower

Many AWS customers have adopted the AWS Landing Zone solution (ALZ) to set up a secure, compliant, multi-account AWS environment. To reduce the burden of managing a landing zone, AWS created the managed service called AWS Control Tower.

No additional features are scheduled for ALZ; it is in long-term support only. Therefore, we recommend that you move to the AWS Control Tower service from ALZ. The blog that is linked in this chapter walks you through different considerations for that move, and it explains how you can plan a successful migration from ALZ to AWS Control Tower.

Blog: Migrate AWS Landing Zone solution to AWS Control Tower

AWS Prescriptive Guidance offers more extensive documentation, including steps for transitioning from ALZ to AWS Control Tower. Essentially, you will enable AWS Control Tower governance in your existing organization that is running ALZ, based upon a number of prerequisites. For information, see Transitioning from AWS Landing Zone to AWS Control Tower.

Walkthrough: Automate Account Provisioning in AWS Control Tower by Service Catalog APIs

AWS Control Tower is integrated with several other AWS services, such as AWS Service Catalog. You can use the APIs to create and provision your member accounts in AWS Control Tower.

The video shows you how to provision accounts in an automated, batch fashion, by calling the AWS Service Catalog APIs. For provisioning, you'll call the `ProvisionProduct` API from the AWS command line interface (CLI), and you'll specify a JSON file that contains the parameters for each account you'd like to set up. The video illustrates installing and using the AWS Cloud9 development environment to perform this work. The CLI commands would be the same if you use AWS Cloudshell instead of AWS Cloud9.

Note

You also can adapt this approach for automating account updates, by calling the `UpdateProvisionedProduct` API of AWS Service Catalog for each account. You can write a script to update the accounts, one by one.
As a completely different automation method, if you are familiar with Terraform, you can provision accounts with AWS Control Tower Account Factory for Terraform (AFT) (p. 139).

**Sample automation administration role**

Here is a sample template you can use to help configure your automation administration role in the management account. You would configure this role in your management account so it can perform the automation with Administrator access in the target accounts.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure the SampleAutoAdminRole

Resources:
  AdministrationRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: SampleAutoAdminRole
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service: cloudformation.amazonaws.com
            Action:
              - sts:AssumeRole
            Path: /
        Policies:
          - PolicyName: AssumeSampleAutoAdminRole
            PolicyDocument:
              Version: 2012-10-17
              Statement:
                - Effect: Allow
                  Action:
                    - sts:AssumeRole
                  Resource:
                    - "arn:aws:iam::*:role/SampleAutomationExecutionRole"
```

**Sample automation execution role**

Here is a sample template you can use to help you set up your automation execution role. You would configure this role in the target accounts.

```
AWSTemplateFormatVersion: "2010-09-09"
Description: "Create automation execution role for creating Sample Additional Role."

Parameters:
  AdminAccountId:
    Type: "String"
    Description: "Account ID for the administrator account (typically management, security or shared services)."
  AdminRoleName:
    Type: "String"
    Description: "Role name for automation administrator access."
    Default: "SampleAutomationAdministrationRole"
  ExecutionRoleName:
    Type: "String"
    Description: "Role name for automation execution."
    Default: "SampleAutomationExecutionRole"
  SessionDurationInSecs:
    Type: "Number"
    Description: "Maximum session duration in seconds."
    Default: 14400
```
After configuring these roles, you call the AWS Service Catalog APIs to perform the automated tasks. The CLI commands are given in the video.

### Sample provisioning input for Service Catalog API

Here is a sample of the input you can give to the Service Catalog ProvisionProduct API if you're using the API to provision AWS Control Tower accounts:

```json
{
  pathId: "lpv2-7n2o3nud1j46e",
  productId: "prod-y422ydgjge2rs",
  provisioningProductName: "Example product 1",
  provisioningArtifactId: "pa-2mmz36cfpj2p4",
  provisioningParameters: [
    {
      key: "AccountEmail",
      value: "abc@amazon.com"
    },
    {
      key: "AccountName",
      value: "ABC"
    },
    {
      key: "ManagedOrganizationalUnit",
      value: "Custom (ou-xfe5-a8hb8ml8)"
    },
    {
      key: "SSOUserEmail",
      value: "abc@amazon.com"
    },
    {
      key: "SSOUserFirstName",
      value: "John"
    },
    {
      key: "SSOUserLastName",
      value: "Smith"
    }
  ],
  provisionToken: "c3c795a1-9824-4fb2-a4c2-4b1841be4068"
}
```
For more information, see the API reference for Service Catalog.

**Note**
Notice that the format of the input string for the value of ManagedOrganizationalUnit has changed from OU_NAME to OU_NAME (OU_ID). The video that follows does not mention this change.

**Video Walkthrough**

This video (6:58) describes how to automate account deployments in AWS Control Tower. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

[Video Walkthrough of Automated Account Provisioning in AWS Control Tower](#)

### Walkthrough: Configure AWS Control Tower Without a VPC

This topic walks through how to configure your AWS Control Tower accounts without a VPC.

If your workload does not require a VPC, you can do the following:

- You can delete the AWS Control Tower virtual private cloud (VPC). This VPC was created when you set up your landing zone.
- You can change your Account Factory settings so that new AWS Control Tower accounts are created without an associated VPC.

**Important**
If you provision Account Factory accounts with VPC internet access settings enabled, that Account Factory setting overrides the control Disallow internet access for an Amazon VPC instance managed by a customer (p. 1346). To avoid enabling internet access for newly provisioned accounts, you must change the setting in Account Factory.

### Delete the AWS Control Tower VPC

Outside of AWS Control Tower, every AWS customer has a default VPC, which you can view on the Amazon Virtual Private Cloud (Amazon VPC) console at [https://console.aws.amazon.com/vpc/](https://console.aws.amazon.com/vpc/). You'll recognize the default VPC, because its name always includes the word *(default)* at the end of the name.

When you set up a AWS Control Tower landing zone, AWS Control Tower deletes your AWS default VPC and creates a new AWS Control Tower default VPC. The new VPC is associated with your AWS Control Tower management account. This topic refers to that new VPC as the *Control Tower VPC*.

When you view your AWS Control Tower VPC in the Amazon VPC console, you will not see the word *(default)* at the end of the name. If you have more than one VPC, you must use the assigned CIDR range to identify the correct AWS Control Tower VPC.

You can delete the AWS Control Tower VPC, but if you later need a VPC in AWS Control Tower, you must create it yourself.

**To delete the AWS Control Tower VPC**

1. Open the Amazon VPC console at [https://console.aws.amazon.com/vpc/](https://console.aws.amazon.com/vpc/).
2. Search for VPC or select VPC from the Service Catalog options. You then see the VPC Dashboard.
3. From the menu on the left, choose Your VPCs. You then see a list of all your VPCs.
4. Identify the AWS Control Tower VPC by its CIDR range.
5. To delete the VPC, choose Actions and then choose Delete VPC.

An AWS (default) VPC already exists in every Region for the AWS Control Tower management account. To follow security best practices, if you choose to delete the AWS Control Tower VPC, it's best also to delete the AWS default VPC associated with the management account from all AWS Regions. Therefore, to secure the management account, remove the default VPC from each Region, as well as removing the VPC created by Control Tower in your AWS Control Tower home region.

Create an Account in AWS Control Tower Without a VPC

If your end-user workloads do not require VPCs, you can use this method to set up end-user accounts that don't have VPCs created for them automatically.

From the AWS Control Tower dashboard, you can view and edit your network configurations settings. After you change the settings so that AWS Control Tower accounts are created without an associated VPC, all new accounts are created without a VPC until you change the settings again.

To configure Account Factory for creating accounts without VPCs

2. Choose Account Factory from the menu on the left.
3. You then see the Account Factory page with the Network Configuration section.
4. Note the current settings if you intend to restore them later.
5. Choose the Edit button in the Network Configuration section.
6. In the Edit account factory network configuration page, go to the VPC Configuration options for new accounts section.

You can follow Option 1 or Option 2, or both, to ensure that AWS Control Tower does not create a VPC when provisioning an account.

a. Option 1 – Removing subnets
   • Turn off the Internet-accessible subnet toggle switch.
   • Set the Maximum number of private subnets value to 0.

b. Option 2 – Removing AWS Regions
   • Clear every checkbox in the Regions for VPC creation column.

7. Choose Save.

Possible Errors

Be aware of these possible errors that could occur when you delete your AWS Control Tower VPC or reconfigure Account Factory to create accounts without VPCs.

• Your existing management account may have dependencies or resources in the AWS Control Tower VPC, which can cause a deletion failure error.
• If you leave the default CIDR in place when setting up to launch new accounts without a VPC, your request fails with an error that the CIDR is not valid.
Walkthrough: Set Up Security Groups in AWS Control Tower With AWS Firewall Manager

The video shows you how to use the AWS Firewall Manager service to provide improvements to your network security for AWS Control Tower. You can designate a security administrator account that's enabled to set up security groups. You will see how you can configure security policies and enforce security rules for your AWS Control Tower organizations, and how you can remediate non-compliant resources by applying policies automatically. You can view the security groups that are in effect for each account and resource (such as an Amazon EC2 instance) in your organization.

You can create your own firewall policies, or you can subscribe to rules from trusted vendors.

Set Up Security Groups With AWS Firewall Manager

This video (8:02) describes how to set up better network infrastructure security for your resources and workloads in AWS Control Tower. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

Video Walkthrough of Firewall Setup in AWS Control Tower.

For more information, see the documentation on how to set up AWS WAF.

Walkthrough: Decommission an AWS Control Tower Landing Zone

AWS Control Tower allows you to set up and govern secure multi-account AWS environments, known as landing zones. The process of cleaning up all of the resources allocated by AWS Control Tower is referred to as decommissioning a landing zone.

If you no longer want to use AWS Control Tower, the automated decommissioning tool cleans up the resources allocated by AWS Control Tower. To begin the automated decommissioning process, navigate to the Landing Zone Settings page, select the decommission tab, and choose Decommission landing zone.

For a list of actions performed during decommissioning, see Overview of the decommissioning process (p. 1435).

Warning
Manually deleting all of your AWS Control Tower resources is not the same as decommissioning. It will not allow you to set up a new landing zone.

Your data and your existing AWS Organizations are not changed by the decommissioning process, in the following ways.

- AWS Control Tower does not remove your data, it only removes parts of the landing zone that it created.
- After the decommissioning process is complete, a few resource artifacts remain, such as Amazon S3 buckets and Amazon CloudWatch Logs log groups. These resources must be deleted manually before you set up another landing zone, and to avoid possible costs associated with maintaining certain resources.
- You can't use automated decommissioning to remove a landing zone that's partially set up. If your landing zone setup process fails, you must resolve the failure state and set it up all the way to make automated decommissioning possible, or you must manually delete the resources individually.
Decommissioning a landing zone is a process with significant consequences, and it cannot be undone. The decommissioning actions taken by AWS Control Tower and the artifacts that remain after decommissioning are described in the following sections.

**Important**
We strongly recommend that you perform this decommissioning process only if you intend to stop using your landing zone. It is not possible to re-create your existing landing zone after you've decommissioned it.

### Overview of the decommissioning process

When you request decommissioning of your landing zone, AWS Control Tower does the following actions.

- Disables each detective control enabled in the landing zone. AWS Control Tower deletes the AWS CloudFormation resources supporting the control.
- Disables each preventive control by removing service control policies (SCPs) from AWS Organizations. If a policy is empty (which it should be after removing all SCPs managed by AWS Control Tower), AWS Control Tower detaches and deletes the policy entirely.
- Deletes all blueprints deployed as AWS CloudFormation StackSets.
- Deletes all blueprints deployed as CloudFormation Stacks across all Regions.
- For each provisioned account, AWS Control Tower does the following actions during the decommissioning process.
  - Deletes records of each account factory account.
  - Revokes the AWS Control Tower permissions to the account by removing the IAM role that AWS Control Tower created (unless additional policies have been added to it) and recreates the standard OrganizationsFullAccessRole IAM role.
  - Removes records of the account from AWS Service Catalog.
  - Removes the account factory product and portfolio from AWS Service Catalog.
  - Deletes the blueprints for the shared (Audit and Log Archive) accounts.
  - Revokes the AWS Control Tower permissions from the shared accounts by removing the IAM role that AWS Control Tower created (unless additional policies have been added to it) and recreates the OrganizationsFullAccessRole IAM role.
  - Deletes records related to the shared accounts.
  - Deletes records related to customer-created OUs.
  - Deletes internal records that identify the home Region.

**Note**
After decommissioning, you may wish to remove the Account Factory VPC blueprint (BP_ACCOUNT_FACTORY_VPC) to clean up the routes and NAT gateways, if your VPC was not empty.

### Resources not removed during decommissioning

Decommissioning a landing zone does not fully reverse the AWS Control Tower setup process. Certain resources remain, which may be removed manually.

**AWS Organizations**

For customers without existing AWS Organizations organizations, AWS Control Tower sets up an organization with two organizational units (OUs), named Security and Sandbox. When you decommission your landing zone, the hierarchy of the organization is preserved, as follows:
• Organizational Units (OUs) you created from the AWS Control Tower console are not removed.
• The Security and Sandbox OUs are not removed.
• The organization is not deleted from AWS Organizations.
• No accounts in AWS Organizations (shared, provisioned, or management) are moved or removed.

AWS IAM Identity Center (SSO)

For customers without an existing IAM Identity Center directory, AWS Control Tower sets up IAM Identity Center and configures an initial directory. When you decommission your landing zone, AWS Control Tower makes no changes to IAM Identity Center. If needed, you can delete the IAM Identity Center information stored in your management account manually. In particular, these areas are unchanged by decommissioning:

• Users created with Account Factory are not removed.
• Groups created by AWS Control Tower setup are not removed.
• Permission sets created by AWS Control Tower are not removed.
• Associations between AWS accounts and IAM Identity Center permission sets are not removed.
• IAM Identity Center directories are not changed.

Amazon S3 Buckets

During setup, AWS Control Tower creates buckets in the logging account for logging and for logging access. When you decommission your landing zone, the following resources are not removed:

• Logging and logging access S3 buckets in the logging account are not removed.
• Contents of the logging and logging access buckets are not removed.

Shared Accounts

Two shared accounts (Audit and Log Archive) are created in the Security OU during AWS Control Tower setup. When you decommission your landing zone:

• Shared accounts that were created during AWS Control Tower setup are not closed.
• The OrganizationAccountAccessRole IAM role is recreated to align with standard AWS Organizations configuration.
• The AWSControlTowerExecution role is removed.

Provisioned Accounts

AWS Control Tower customers can use account factory to create new AWS accounts. When you decommission your landing zone:

• Provisioned accounts you created with Account Factory are not closed.
• Provisioned products in AWS Service Catalog are not removed. If you clean those up by terminating them, their accounts are moved into the Root OU.
• The VPC that AWS Control Tower created is not removed, and the associated AWS CloudFormation stack set (BP_ACCOUNT_FACTORY_VPC) is not removed.
• The OrganizationAccountAccessRole IAM role is recreated to align with standard AWS Organizations configuration.
• The AWSControlTowerExecution role is removed.
Resources not removed during decommissioning

CloudWatch Logs Log Group

A CloudWatch Logs log group, aws-controltowertower/CloudTrailLogs, is created as part of the blueprint named AWSControlTowerBP-BASELINE-CLOUDTRAIL-MANAGEMENT. This log group is not removed. Instead, the blueprint is deleted and the resources are retained.

- This log group must be deleted manually before you set up another landing zone.

Note

Customers on landing zone 3.0 and later do not need to delete their individual enrolled account's CloudTrail logs and CloudTrail logs roles, because these are created in the management account only, for the organization-level trail. Beginning with landing zone version 3.2, AWS Control Tower creates an EventBridge rule, called AWSControlTowerManagedRule. This rule is created in each member account, for all governed Regions. The rule is not deleted automatically during decommissioning, so you must delete it manually from the shared and member accounts for all governed Regions before you can set up a landing zone in a new Region.

Procedures for how to delete AWS Control Tower resources are given in Manage AWS Control Tower Resources (p. 1437).

Manage AWS Control Tower Resources

This document provides instructions for how to remove AWS Control Tower resources individually, as part of regular maintenance and administrative tasks. The procedures given in this chapter are intended only for removing individual resources, or a few resources, when needed. It not the same as decommissioning your landing zone.

Two types of tasks may require you to remove resources:

- To delete resources as you manage your landing zone in ordinary situations.
- To clean up resources that remain after automated decommissioning.

Warning

Manually removing resources will not allow you to set up a new landing zone. It is not the same as decommissioning. If you intend to decommission your AWS Control Tower landing zone, follow the instructions on Walkthrough: Decommission an AWS Control Tower Landing Zone (p. 1434) before you take any actions described in this chapter. The instructions in this chapter can help you clean up resources that remain after automated decommissioning is complete. Even if you delete all of your landing zone resources manually, it is not the same as decommissioning the landing zone, and you may incur unexpected charges.

If you need to remove an account from AWS Control Tower, see the following sections to close an account:

- Unmanage an account
- Close an account created in Account Factory

Do I need decommissioning instead of deleting?

If you no longer intend to use AWS Control Tower for your enterprise, or if you require a major redeployment of your organizational resources, you may want to decommission the resources created when you initially set up your landing zone.

- After the decommissioning process is complete, a few resource artifacts remain, such as Amazon S3 buckets and Amazon CloudWatch Logs log groups.
• You must clean up the remaining resources in your accounts manually before you set up another landing zone, and to avoid the possibility of unexpected charges. For more information, see Resources not removed during decommissioning (p. 1435).

**Warning**
We strongly recommend that you perform a decommissioning process only if you intend to stop using your landing zone. This process cannot be undone.

**About removing AWS Control Tower resources**

The individual procedures in this chapter guide you through manual methods of removing AWS Control Tower resources. These procedures can be followed when you need to delete a specific resource from your landing zone.

Before performing these procedures, unless it's otherwise indicated, you must be signed in to the AWS Management Console in the home Region for your landing zone, and you must be signed in as an IAM user or user in IAM Identity Center with administrative permissions for the management account that contains your landing zone.

**Warning**
These are destructive actions that can introduce governance drift into your AWS Control Tower setup. They cannot be undone.

**Topics**

- **Delete SCPs** (p. 1438)
- **Delete StackSets and Stacks** (p. 1438)
- **Delete Amazon S3 Buckets in the Log Archive Account** (p. 1439)
- **Remove an Account Factory Portfolio and Product** (p. 1440)
- **Remove AWS Control Tower Roles and Policies** (p. 1441)
- **AWS Control Tower resource help** (p. 1442)

**Delete SCPs**

AWS Control Tower uses service control policies (SCPs) for its controls. This procedure walks through how to delete the SCPs specifically related to AWS Control Tower.

**To delete AWS Organizations SCPs**

2. Open the **Policies** tab, and find the Service Control Policies (SCPs) that have the prefix **aws-guardrails-** and do the following for each SCP:
   a. Detach the SCP from the associated OU.
   b. Delete the SCP.

**Delete StackSets and Stacks**

AWS Control Tower uses StackSets and stacks to deploy AWS Config Rules related to controls in your landing zone. The following procedures walk through how to delete these specific resources.

**To delete AWS CloudFormation StackSets**

2. From the left navigation menu, choose StackSets.
3. For each StackSet with the prefix AWSControlTower, do the following. If you have many accounts in a StackSet, this can take some time.
   a. Choose the specific StackSet from the table in the dashboard. This opens the properties page for that StackSet.
   b. At the bottom of the page, in the Stacks table, make a record of the AWS account IDs for all the accounts in the table. Copy the list of all accounts.
   c. From Actions, choose Delete stacks from StackSet.
   d. On Set deployment options, from Deployment locations, choose Deploy stacks in accounts.
   e. In the text field, enter the AWS account IDs you made a record of in step 3.b, separated by commas. For example: 123456789012, 098765431098, and so on.
   f. From Specify regions, choose Add all, leave the rest of the parameters on the page set to their defaults, and choose Next.
   g. On the Review page, review your choices, and then choose Delete stacks.
   h. On the StackSet properties page, you can begin this procedure again for your other StackSets.
4. The process is complete when the records in the Stacks table of the different StackSets properties pages are empty.
5. When the records in the Stacks table are empty, choose Delete StackSet.

To delete AWS CloudFormation stacks

2. From the Stacks dashboard, search for all of the stacks with the prefix AWSControlTower.
3. For each stack in the table, do the following:
   a. Choose the check box next to the name of the stack.
   b. From the Actions menu, choose Delete Stack.
   c. In the dialog box that opens, review the information to make sure it's accurate, and choose Yes, Delete.

Delete Amazon S3 Buckets in the Log Archive Account

The following procedures guide you through how to sign in to the log archive account as an IAM Identity Center user in the AWSControlTowerExecution group and then delete the Amazon S3 buckets in your log archive account.

To sign in to your log archive account with the right permissions

1. Open the Organizations console at https://console.aws.amazon.com/organizations/.
2. From the Accounts tab, find the Log archive account.
3. From the right pane that opens, make a record of the log archive account number.
4. From the navigation bar, choose your account name to open your account menu.
5. Choose Switch Role.
6. On the page that opens, provide the account number for the log archive account in Account.
7. For Role, enter AWSControlTowerExecution.
8. The Display Name populates with text.
9. Choose your favorite Color.
10. Choose Switch Role.
To delete Amazon S3 buckets
1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Search for bucket names that contain `aws-controltower`.
3. For each bucket in the table, do the following:
   a. Choose the check box for the bucket in the table.
   b. Choose Delete.
   c. In the dialog box that opens, review the information to make sure it's accurate, enter the name of the bucket to confirm, and then choose Confirm.

Remove an Account Factory Portfolio and Product

The following procedure guides you through how to sign in as an IAM Identity Center user in the AWSServiceCatalogAdmins group and then clean up your Account Factory portfolio and products.

To sign in to your management account with the right permissions
1. Go to your user portal URL at directory-id.awsapps.com/start
2. From AWS Account, find the Management account.
3. From AWSServiceCatalogAdminFullAccess, choose Management console to sign in to the AWS Management Console as this role.

To clean up Account Factory
1. Open the Service Catalog console at https://console.aws.amazon.com/servicecatalog/.
2. From the left navigation menu, choose Portfolios list.
3. In the Local Portfolios table, search for a portfolio named AWS Control Tower Account Factory Portfolio.
4. Choose the name of that portfolio to go to its details page.
5. Expand the Constraints section of the page, and choose the radio button for the constraint with the product name AWS Control Tower Account Factory.
6. Choose REMOVE CONSTRAINTS.
7. In the dialog box that opens, review the information to make sure it's accurate, and then choose CONTINUE.
8. From the Products section of the page, choose the radio button for the product named AWS Control Tower Account Factory.
9. Choose REMOVE PRODUCT.
10. In the dialog box that opens, review the information to make sure it's accurate, and then choose CONTINUE.
11. Expand the Users, Groups, and Roles section of the page, and choose the check boxes for all the records in this table.
12. Choose REMOVE USERS, GROUP OR ROLE.
13. In the dialog box that opens, review the information to make sure it's accurate, and then choose CONTINUE.
14. From the left navigation menu, choose Portfolios list.
15. In the Local Portfolios table, search for a portfolio named AWS Control Tower Account Factory Portfolio.
16. Choose the radio button for that portfolio, and then choose DELETE PORTFOLIO.
17. In the dialog box that opens, review the information to make sure it's accurate, and then choose **CONTINUE**.
18. From the left navigation menu, choose **Product list**.
19. On the **Admin products** page, search for the product named **AWS Control Tower Account Factory**.
20. Choose the product to open the **Admin product details** page.
21. From **Actions**, choose **Delete product**.
22. In the dialog box that opens, review the information to make sure it's accurate, and then choose **CONTINUE**.

### Remove AWS Control Tower Roles and Policies

These procedures walk you through how to clean up the roles and policies that AWS Control Tower created when your landing zone was set up, or later.

#### To delete the IAM Identity Center AWSServiceCatalogEndUserAccess role

1. Open the AWS IAM Identity Center console at [https://console.aws.amazon.com/singlesignon/](https://console.aws.amazon.com/singlesignon/).
2. Change your AWS Region to your home Region, which is the Region where you initially set up AWS Control Tower.
3. From the left navigation menu, choose **AWS accounts**.
4. Choose your management account link.
5. Choose the dropdown for **Permission sets**, select **AWSServiceCatalogEndUserAccess**, and then choose **Remove**.
6. Choose **AWS accounts** from the left panel.
7. Open the **Permission sets** tab.
8. Select **AWSServiceCatalogEndUserAccess** and delete it.

#### To delete IAM roles

2. From the left navigation menu, choose **Roles**.
3. From the table, search for roles with the name **AWSControlTower**.
4. For each role in the table, do the following:
   a. Choose the check box for the role.
   b. Choose **Delete role**.
   c. In the dialog box that opens, review the information to make sure it's accurate, and then choose **Yes, delete**.

#### To delete IAM policies

2. From the left navigation menu, choose **Policies**.
3. From the table, search for policies with the name **AWSControlTower**.
4. For each policy in the table, do the following:
   a. Choose the check box for the policy.
   b. Choose **Policy actions**, and **Delete** from the dropdown menu.
   c. In the dialog box that opens, review the information to make sure it's accurate, and then choose **Delete**.
AWS Control Tower resource help

If you encounter any issues that you can't resolve when you remove AWS Control Tower resources, contact AWS Support.

How to decommission a landing zone

To decommission your AWS Control Tower landing zone, follow the procedure given here.

Note
We recommend that you unmanage your enrolled accounts prior to decommissioning.

1. Navigate to the Landing Zone Settings page in the AWS Control Tower console.
2. Choose Decommission your landing zone within the Decommission your landing zone section.
3. A dialog appears, explaining the action you are about to perform, with a required confirmation process. To confirm your intent to decommission, you must select every box and type the confirmation as requested.

   Important
   The decommissioning process cannot be undone.

4. If you confirm your intent to decommission your landing zone, you are redirected to the AWS Control Tower home page while decommissioning is in progress. The process may require up to two hours.
5. When decommissioning has succeeded, you must delete remaining resources manually before setting up a new landing zone from the AWS Control Tower console. These remaining resources include some specific Amazon S3 buckets, organizations, and CloudWatch Logs log groups.

   Note
   These actions may have significant consequences for your billing and compliance activities. For example, failure to delete these resources can result in unexpected charges.

   For more information about how to delete resources manually, see About removing AWS Control Tower resources (p. 1438).

6. If you intend to set up a new landing zone in a new AWS Region, follow this additional step. Enter the following command through the CLI:

   ```bash
   aws organizations disable-aws-service-access --service-principal controltower.amazonaws.com
   ```

Manual cleanup tasks required after decommissioning

- You must specify different email addresses for the Log archive and Audit accounts if you create a new landing zone after decommissioning one, or follow the procedure for bringing your own existing Log archive or Audit accounts.
- The CloudWatch Logs log group, aws-controltower/CloudTrailLogs, must be deleted manually before you set up another landing zone.
- The two Amazon S3 buckets with reserved names for logs must be removed, or renamed, manually.
- You must delete, or rename, the existing Security and Sandbox organizational units manually.

   Note
   Before you can delete the AWS Control Tower Security OU organization, you must first delete the logging and audit accounts, but not the management account. To delete these accounts, you must When to sign in as a root user (p. 41) to the audit account and to the logging account and delete them individually.
Setup after decommissioning a landing zone

After you decommission your landing zone, you cannot successfully execute setup again until manual cleanup is complete. Also, without manual cleanup of these remaining resources, you may incur unexpected billing charges. You must attend to these issues:

- The AWS Control Tower management account is part of the AWS Control Tower Root OU. Be sure that these IAM roles and IAM policies are removed from the management account:
  - Roles:
    - AWSControlTowerAdmin
    - AWSControlTowerCloudTrailRole
    - AWSControlTowerStackSetRole
  - Policies:
    - AWSControlTowerAdminPolicy
    - AWSControlTowerCloudTrailRolePolicy
    - AWSControlTowerStackSetRolePolicy
- You may wish to delete or update the existing IAM Identity Center configuration for AWS Control Tower before you set up a landing zone again, but it is not required that you delete it.
- You may wish to remove the VPC created by AWS Control Tower.
- Setup fails if the email addresses specified for the logging or audit accounts are associated with an existing AWS account. You may close the AWS accounts, or use different email addresses to set up a landing zone again. Alternatively, you may re-use these existing shared accounts, with the feature that allows you to bring your own logging and audit accounts. For more information, see Considerations for bringing existing security or logging accounts (p. 106).
- Setup fails if Amazon S3 buckets with the following reserved names already exist in the logging account:
  - `aws-controltower-logs-{accountId}-{region}` (used for the logging bucket).
  - `aws-controltower-s3-access-logs-{accountId}-{region}` (used for the logging access bucket).

You must either rename or remove these buckets, or use a different account for the logging account.
- Setup fails if the management account has the existing log group, `aws-controltower/CloudTrailLogs`, in CloudWatch Logs. You must either rename or remove the log group.
Before you set up in a new AWS Region

If you intend to set up a new landing zone in a new AWS Region, follow these additional steps.

- Enter the following command through the CLI:

  ```bash
  aws organizations disable-aws-service-access --service-principal controltower.amazonaws.com
  ```

- Delete the remaining managed rule, called AWSControlTowerManagedRule, from shared and member accounts for all governed Regions.

  **Note**
  You cannot set up a new landing zone in an organization with top-level OUs named either **Security** or **Sandbox**. You must rename or remove these OUs to set up a landing zone again.
Troubleshooting

If you encounter issues while using AWS Control Tower, you can use the following information to resolve them according to our best practices. If the issues you encounter are outside the scope of the following information, or if they persist after you've tried to resolve them, contact AWS Support.

Landing Zone Launch Failed

Common causes of landing zone launch failure:

- Lack of response to a confirmation email message.
- AWS CloudFormation StackSet failure.

**Confirmation email messages:** If your management account is less than an hour old, you may encounter issues when the additional accounts are created.

**Action to take**

If you encounter this issue, check your email. You might have been sent confirmation email that is awaiting response. Alternatively, we recommend that you wait an hour, and then try again. If the issue persists, contact AWS Support.

**Failed StackSets:** Another possible cause of landing zone launch failure is AWS CloudFormation StackSet failure. AWS Security Token Service (STS) regions must be enabled in the management account for all AWS Regions that AWS Control Tower is governing, so that the provisioning can be successful; otherwise, stack sets will fail to launch.

**Action to take**

Be sure to enable all of your required AWS Security Token Service (STS) endpoint regions before you launch AWS Control Tower.

To view a list of AWS Regions that AWS Control Tower supports, see How AWS Regions Work With AWS Control Tower (p. 97).

Landing zone not up to date error

If you have not updated your landing zone recently, you may receive an error when you try to regain access to AWS Control Tower. You may see an error message similar to this one:

**Unable to access Control Tower**

*Your account has been inactive for too long. Due to inactivity, you must update your landing zone for access to AWS Control Tower.*

However, your landing zone update may fail.

**Steps to take**

Sign in to the management account of your organization, and sign in as root user. Your IAM user or user in IAM Identity Center must have AWS Control Tower administrator permissions and be part of the **AWSControlTowerAdmins** group. Then try the update again.
New Account Provisioning Failed

If you encounter this issue, check for these common causes.

When you filled out the account provisioning form, you may have:

- specified tagOptions,
- enabled SNS notifications,
- enabled provisioned product notifications.

Try again to provision your account, without specifying any of those options. For more information, see Provision accounts with AWS Service Catalog Account Factory (p. 122).

Other common causes for failure:

- If you created a provisioned product plan (to view resource changes), your account provisioning may remain in an In progress state indefinitely.
- Creation of a new account in Account Factory will fail while other AWS Control Tower configuration changes are in progress. For example, while a process is running to add a control to an OU, Account Factory will display an error message if you try to provision an account.

To check the status of a previous action in AWS Control Tower

- Navigate to AWS CloudFormation > StackSets
- Check each stack set related to AWS Control Tower (prefix: "AWSControlTower")
- Look for AWS CloudFormation StackSets operations that are still running.

If your account provisioning takes longer than one hour, it's best to terminate the provisioning process and try again.

Failed to Enroll an Existing Account

If you try once to enroll an existing AWS account and that enrollment fails, when you try a second time, the error message may tell you that the stack set exists. To continue, you must remove the provisioned product in Account Factory.

If the reason for the first enrollment failure was that you forgot to create the AWSControlTowerExecution role in the account in advance, the error message you'll receive correctly tells you to create the role. However, when you try to create the role, you are likely to receive another error message stating that AWS Control Tower could not create the role. This error occurs because the process has been partially completed.

In this case, you must take two recovery steps before you can proceed with enrolling your existing account. First, you must terminate the Account Factory provisioned product through the AWS Service Catalog console. Next, you must use the AWS Organizations console to manually move the account out of the OU and back to the root. After that is done, create the AWSControlTowerExecution role in the account, and then fill in the Enroll account form again.

Another possible cause of enrollment failure is that the account has existing AWS Config resources. In that case, see Enroll accounts that have existing AWS Config resources for instructions on how you can modify your existing resources.
Unable to Update an Account Factory Account

When an account is in an inconsistent state, it cannot be updated successfully from Account Factory or AWS Service Catalog.

Case 1: You may encounter an error message similar to this one:

AWS Control Tower could not baseline VPC in the managed account because of existing resource dependencies.

Common cause: AWS Control Tower always removes the AWS default VPC during initial provisioning. To have an AWS default VPC in an account, you must add it after account creation. AWS Control Tower has its own default VPC that replaces the AWS default VPC, unless you set up Account Factory the way the walkthrough shows you—so that AWS Control Tower doesn't provision a VPC at all. Then the account has no VPC. You'd have to re-add the AWS default VPC if you want to use that one.

However, AWS Control Tower doesn't support the AWS default VPC. Deploying one causes the account to enter a Tainted state. When it is in that state, you cannot update the account through AWS Service Catalog.

Action to take: You must delete the default VPC that you added, and then you will be able to update the account.

Note
The Tainted state causes a follow-on issue: An account that is not updated may prevent enabling controls on the OU of which it is a part.

Case 2: You may see an error message similar to this one:

AWS Control Tower detects that your enrolled account has been moved to a new organizational unit.

Common cause: You attempted to move an account from one registered OU to another, but old AWS Config rules remain. The account is in an inconsistent state.

Action to take:

If the account move was intended:

- Terminate the account in Service Catalog.
- Enroll it again.
- Context/impact: Deployed AWS Config rules don't match the configuration dictated by the destination OU.
- AWS Config rules may remain from the previous OU, causing unintended spending.
- Attempts to re-enroll or update the account will fail due to resource naming conflicts.

If the account move was unintended:

- Return the account to its original OU.
- Update the account from Service Catalog.
- In the launch parameters, enter the OU that the account was originally in.
- Context/impact: If the account is not returned to its original OU, its state will be inconsistent with the controls dictated by the new OU it's in.
- Updating an account is not a valid remediation, because it does not delete the AWS Config rules associated with its previous OU.
Unable to Update Landing Zone

When an account is in a Closed or Suspended state, you may encounter an issue when you try to update your landing zone. You must delete the provisioned product on every closed account before you perform an update to the landing zone.

On the AWS Service Catalog provisioned product page, you may see an error message similar to this one:

AWSControlTowerExecution role can’t be assumed on the account.

Common cause: You have suspended an account without deleting the provisioned product.

Action to take: If you see this error, you have two options:

1. Contact AWS Support and reopen the account, delete the provisioned product, then close the account again.
2. Remove the resources from the StackSets that have been orphaned because of the account closure. (This option is available only if the StackSets have instances in Current state that you are not removing.)

To remove the resources from the StackSets, do this for each closed account:

- Go into each of the AWS Control Tower StackSets and remove the StackInstances from every region, for the account that has been closed.
- IMPORTANT: Choose the Retain Stack option so the StackSet removes only the stack instances. StackSet can't assume a role from the closed account, so it will fail if it tries to assume the AWSControlTowerExecution role, which leads to the error message you received.

Failure Error that Mentions AWS Config

If AWS Config is enabled in any AWS Region supported by AWS Control Tower, you may receive an error message because a pre-check has failed. The message might not seem to explain the problem adequately, due to some underlying behavior of AWS Config.

You may receive an error message, similar to one of these:

- AWS Control Tower cannot create an AWS Config delivery channel because one already exists. To continue, delete the existing delivery channel and try again.
- AWS Control Tower cannot create an AWS Config configuration recorder because one already exists. To continue, delete the existing delivery channel and try again.

Common cause: When the AWS Config service is enabled on an AWS account, it creates a configuration recorder and delivery channel with a default naming. If you disable the AWS Config service through the console, it does not delete the configuration recorder or the delivery channel. You must delete them through the CLI, or modify them for AWS Control Tower use. If the AWS Config service is enabled in any one of the Regions supported by AWS Control Tower, it can result in this failure.

If the account has existing AWS Config resources, see Enroll accounts that have existing AWS Config resources for instructions on how you can modify your existing resources.
Action to take: Delete the configuration recorder and delivery channel in all supported regions. Disabling AWS Config is not enough, the configuration recorder and delivery channel must be deleted by means of the CLI. After you’ve deleted the configuration recorder and delivery channel from the CLI, you can try again to launch AWS Control Tower and enroll the account.

If you are in the process of deploying a provisioned product, you must delete the provisioned product before you retry. Otherwise, you may see an error message similar to this one:

- An error occurred (InvalidParametersException) when calling the ProvisionProduct operation: A stack named Stackname already exists.

In the message, Stackname specifies the name of the stack.

Here are some example AWS Config CLI commands you can use to determine the status of your configuration recorder and delivery channel.

View commands:

- aws configservice describe-delivery-channels
- aws configservice describe-delivery-channel-status
- aws configservice describe-configuration-recorders
- The normal response is something like "name": "default"

Delete commands:

- aws configservice stop-configuration-recorder --configuration-recorder-name NAME-FROM-DESCRIBE-OUTPUT
- aws configservice delete-delivery-channel --delivery-channel-name NAME-FROM-DESCRIBE-OUTPUT
- aws configservice delete-configuration-recorder --configuration-recorder-name NAME-FROM-DESCRIBE-OUTPUT

For more information, see the AWS Config documentation

- Managing the Configuration Recorder (AWS CLI)
- Managing the Delivery Channel

No Launch Paths Found Error

When you're trying to create a new account, you may see an error message similar to this one:

No launch paths found for resource: prod-dpqfywxxxx

This error message is generated by AWS Service Catalog, which is the integrated service that helps provision accounts in AWS Control Tower.

Common Causes:

- You may be logged in as root. AWS Control Tower does not support creating accounts when you're logged in as root user.
- Your IAM Identity Center user has not been added to the appropriate permission group. You may need to add your IAM Identity Center user to one of these permission groups: AWSAccountFactory (for end-user access) or AWSServiceCatalogAdmins (for admin access).
• If you are authenticated as an IAM user, you must add it to the AWS Service Catalog portfolio so that it has the correct permissions.
• This issue also occurs if you have the correct permissions, but AWS Control Tower drift is detected and a repair is necessary.

Received an Insufficient Permissions Error

It's possible that your account may not have the necessary permissions to perform certain work in certain AWS Organizations. If you encounter the following type of error, check all the permissions areas, such as IAM or IAM Identity Center permissions, to make sure your permission is not being denied from those places:

You have insufficient permissions to perform AWS Organizations API actions.

If you believe your work requires the action you're attempting, and you can't locate any relevant restriction, contact your system administrator or AWS Support.

Detective controls are not taking effect on accounts

If you've recently expanded your AWS Control Tower deployment into a new AWS Region, newly-applied detective controls do not take effect on new accounts you create in any Region until the individual accounts within OUs governed by AWS Control Tower are updated. Existing detective controls on existing accounts are still in effect.

If you try to enable a detective control before updating your accounts, you may see an error message similar to this one:

AWS Control Tower can't enable the selected control on this OU. AWS Control Tower cannot apply the control on the OU ou-xxx-xxxxxxxx, because child accounts have dependencies that are missing. Update all child accounts under the OU, then try again.

Action to take: Update accounts.

To update your accounts from the AWS Control Tower console, see When to update AWS Control Tower OUs and accounts (p. 194).

To update multiple individual accounts programmatically, you can use the APIs from AWS Service Catalog and the AWS CLI to automate the updates. For more information about how to approach the update process, see this Video Walkthrough (p. 1432). You can substitute the UpdateProvisionedProduct API for the ProvisionProduct API shown in the video.

If you have further difficulties with enabling detective controls on your accounts, contact AWS Support.

Rate exceeded error returned by the AWS Organizations API

Possible cause
Your workload was running while AWS Control Tower was running a daily scan to check whether your SCPs have drifted.

Steps to follow

If you encounter an API throttling or rate exceeded error, try these steps:

- Run your workloads at a different time. (Refer to the AWS Control Tower SCP invariance scan schedule by Region to find out when AWS Control Tower runs its audit scans.)
- If you are calling the APIs directly through HTTP: Use the AWS SDK, which automatically retries failed actions
- Request a limit increase through Service Quotas and AWS Support

An example of troubleshooting instructions for API throttling in Elastic Beanstalk can be found here: https://aws.amazon.com/premiumsupport/knowledge-center/elastic-beanstalk-api-throttling-errors/

Failure to move an Account Factory account directly from one AWS Control Tower landing zone to another AWS Control Tower landing zone

Warning
This practice does not meet the prerequisite for eligible account enrollment, because eligible accounts must be part of the same overall AWS Organization, and each organization may have only one landing zone. If you have tried to do this action and you find yourself receiving multiple error messages, here is some information that might be helpful.

To move an account that you've provisioned through Account Factory into another landing zone that's managed by AWS Control Tower, under another management account, you must remove all of the IAM roles and the stacks associated with that account from the original OU. Remove these resources from every Region in which the account is deployed.

Note
The best way to remove the resources is to deprovision the account in its original OU before you try to move it.

If you don't remove the resources, enrollment into the new OU will fail, somewhat spectacularly. You may encounter one or more error messages, and you will keep receiving similar error messages until the remaining roles and stacks are removed from every Region in which the account was deployed.

Each time you receive an error message, you must remove the account from the new OU, delete the old resource that is the subject of the error message, and then attempt to move the account back into the new OU. This process of removing-and-deleting must be repeated for every remaining resource, for every Region in which the account was deployed, possibly 10 or 20 times. These repeated errors occur because the account was provisioned into an OU with an SCP that prevents IAM role deletion. You can make the recovery process shorter by deleting all the account's resources before you retry.

The examples below represent the types of failure messages you may receive if undeleted roles and stacks remain. You would most likely see one of these messages at a time, for each time you attempt to enroll the account, as long as old resources remain.

The values of the resource ID strings have been modified for the examples. Their values will not be the same in an error message you may receive. You may see a message similar to the following examples:
• AWS Control Tower cannot create the IAM role `aws-controltower-AdministratorExecutionRole` because the role already exists. To continue, delete the existing IAM role and try again.

• AWS Control Tower cannot create the IAM role `aws-controltower-ConfigRecorderRole` because the role already exists. To continue, delete the existing IAM role and try again.

• AWS Control Tower cannot create the IAM role `aws-controltower-ForwardSnsNotificationRole` because the role already exists. To continue, delete the existing IAM role and try again.

Or you may see an error message about a stack set failure, similar to this one:

```
"Error":"StackSetFailState",
"Cause":"StackSetOperation on AWSControlTowerBP-BASELINE-CLOUDWATCH with id 8aXXXXf5-e0XX-4XXa-bc4XX-dXXXXXee31 has reached SUCCEEDED state but has 1 NON-CURRENT stack instances; here is the summary :{
  Stack Set Id: AWSControlTowerBP-BASELINE-CLOUDWATCH:40XXXbf2-Xead-46a1-XXXa-eXXXXecb2ee2,
  Stack instance Id: arn:aws:cloudformation:eu-west-1:1X23456789XX:stack/StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-4feXXXXXX-ecXX-XXc6-bXXX-4ae678/4feXXXXXX-ecX-4ae123458,
  Status: OUTDATED,
  Status Reason: ResourceLogicalId:ForwardSnsNotification,
  ResourceType:AWS::Lambda::Function,
  ResourceStatusReason:aws-controltower-NotificationForwarder already exists in stack
```

After all of the remaining resources are removed from the first OU, you’ll be able to invite, provision, or enroll the account into the new OU successfully.

**AWS Support**

If you want to move your existing member accounts into a different support plan, you can sign in to each account with root account credentials, compare plans, and set the support level that you prefer.

We recommend that you update the MFA and account security contacts when you make changes to your support plan.
Related information

This topic lists common use cases and best practices for AWS Control Tower capabilities and additional enhancements. This topic also includes links to relevant blog posts, technical documentation, and related resources that can help you as you work with AWS Control Tower.

Tutorials and labs

- **AWS Control Tower lab** – These labs provide a high-level overview of common tasks related to AWS Control Tower.
- On the AWS Control Tower dashboard, choose **Get personalized guidance** if you have a use case in mind but you’re not sure where to start.
- Try visiting a [curated list of YouTube videos](#) that explain more about how to use AWS Control Tower functionality.

Networking

Set up repeatable and manageable patterns for networks in AWS. Learn more about design, automation, and appliances that are commonly used by customers.

- **AWS Quick Start VPC Architecture** – This Quick Start guide provides a networking foundation based on AWS best practices for your AWS Cloud infrastructure. It builds an AWS Virtual Private Network environment with public and private subnets where you can launch AWS services and other resources.
- **Self-service VPCs in AWS Control Tower using AWS Service Catalog** – This blog post describes a way to set up Account Factory so you can provision accounts with customized VPCs.
- **Implementing Serverless Transit Network Orchestrator (STNO) in AWS Control Tower** – This blog post demonstrates how to automate network connectivity access across accounts. This blog is intended for AWS Control Tower administrators, or those responsible for managing networks within their AWS environment.

Security, identity, and logging

Extend your security posture, integrate with external or existing identity providers, and centralize logging systems.

**Security**

- **Automating AWS Security Hub Alerts with AWS Control Tower lifecycle events** – This blog post describes how to automate Security Hub enablement and configuration in an AWS Control Tower multi-account environment on existing and new accounts.
- **Enabling AWS Identity and Access Management** – This blog post describes how to enhance your organizational security visibility by enabling and centralizing IAM Access Analyzer findings.
- **AWS Systems Manager Parameter Store** provides secure, hierarchical storage for configuration data management and secrets management. You can use it to share configuration information in a secure location, for use by AWS Systems Manager and by AWS CloudFormation. For example, you can store a list of Regions in which you want to deploy conformance packs.
Identity

- Link Azure AD user identity into AWS accounts and applications for single sign-on – This blog post describes how to use Azure AD with IAM Identity Center and AWS Control Tower.
- Manage access to AWS centrally for Okta users with AWS IAM Identity Center – This blog post describes how to use Okta with IAM Identity Center and AWS Control Tower.

Logging

- AWS Centralized Logging Solution – This solutions post describes the Centralized Logging solution which enables organizations to collect, analyze, and display logs on AWS across multiple accounts and AWS Regions.

Deploying resources and managing workloads

Deploy and manage resources and workloads.

- Getting Started Library integration – This blog post describes Getting Started portfolios you can use.
- Continuous deployment of Cloud Custodian to AWS Control Tower

Working with existing organizations and accounts

Work with existing AWS organizations and accounts.

- Enroll an account – This user guide topic describes how to enroll an existing AWS account in AWS Control Tower.
- Bring an account under AWS Control Tower – This blog post describes how to deploy AWS Control Tower into your existing AWS organizations.
- Extend AWS Control Tower governance using AWS Config conformance packs – This blog post describes how to deploy AWS Config conformance packs to assist with bringing existing accounts and organizations into governance by AWS Control Tower.
- How to Detect and Mitigate Guardrail Violation with AWS Control Tower – This blog post describes how to add controls and how to subscribe to SNS notifications so that you can be notified by email of control compliance violations.

Automation and integration

Automate account creation and integrate lifecycle events with AWS Control Tower.

- Lifecycle events – This blog post describes how to use lifecycle events with AWS Control Tower.
- Automate account creation – This blog post describes how to set up automated account creation in AWS Control Tower.
- Amazon VPC flow log automation – This blog post describes how to automate and centralize Amazon VPC Flow Logs in a multi-account environment.
- Automate VPC tagging with AWS Control Tower lifecycle events – This blog post describes how to automate resource tagging for VPCs, by means of lifecycle events in AWS Control Tower.
- Automated account management – This blog post describes how to automate account management tasks after your AWS Control Tower environment is set up.
Migrating workloads

Use other AWS services with AWS Control Tower to assist in workload migration.

- **CloudEndure migration** – This blog post describes how to combine CloudEndure and other AWS services with AWS Control Tower to assist in workload migration.

Related AWS services

AWS Control Tower acts as an orchestration layer for AWS Organizations. Therefore, by means of the AWS Organizations console and APIs, you have access to over 20 other AWS services that work with AWS Control Tower. These additional services are not accessible directly through the AWS Control Tower console.

- For a full list of services available to AWS Control Tower by means of AWS Organizations, see [AWS services that you can use with AWS Organizations](#).
- To enable multi-account capabilities for these related AWS services, you must enable trusted access. For more information, see [Using AWS Organizations with other AWS services](#).

  **Note**
  
  Remember that AWS IAM Identity Center, AWS Config, and AWS CloudTrail are set up for you in AWS Control Tower and fully integrated. You do not need to modify your trusted access or delegated administration settings for these services.

- Some AWS services available through AWS Organizations can use delegated administration, including AWS Systems Manager and AWS Firewall Manager. For more information, see [Configuring a Delegated Administrator](#) and [Enabling a delegated administrator account for Firewall Manager](#). Also see this video, [Set up security groups with AWS Firewall Manager](#).

AWS Marketplace solutions

Discover solutions from AWS Marketplace.

- **AWS Control Tower Marketplace** – AWS Marketplace offers a broad range of solutions for AWS Control Tower to help you integrate third-party software. These solutions help solve key infrastructure and operational use cases including identity management, security for a multi-account environment, centralized networking, operational intelligence, and security information and event management (SIEM).
AWS Control Tower release notes

Following are details about AWS Control Tower releases that require an update for an AWS Control Tower landing zone, as well as releases that are incorporated into the service automatically.

Features and releases are listed in reverse chronological order (most recent first) based on the date on which they were officially announced to the public. Because there can be a lag between when the feature or release is documented and when it is officially announced, the date listed for a feature or release here may differ slightly from the date in the Document history (p. 1492).

Features released in 2023 (p. 1456)
Features released in 2022 (p. 1469)
Features released in 2021 (p. 1478)
Features released in 2020 (p. 1485)
Features released in 2019 (p. 1489)

January 2023 - Present

Since January 2023, AWS Control Tower has released the following updates:

- AWS Control Tower available in Asia Pacific (Melbourne) Region (p. 1457)
- Transition to new AWS Service Catalog External product type (p. 1457)
- New control API available (p. 1457)
- AWS Control Tower adds additional controls (p. 1458)
- New drift type reported: trusted access disabled (p. 1459)
- Four additional AWS Regions (p. 1460)
- AWS Control Tower available in Tel Aviv Region (p. 1460)
- AWS Control Tower launches 28 new proactive controls (p. 1460)
- AWS Control Tower deprecates two controls (p. 1462)
- AWS Control Tower landing zone version 3.2 (p. 1462)
- AWS Control Tower handles accounts based on ID (p. 1463)
- Additional Security Hub detective controls available in the AWS Control Tower controls library (p. 1463)
- AWS Control Tower publishes control metadata tables (p. 1464)
- Terraform support for Account Factory Customization (p. 1464)
- AWS IAM Identity Center self-management available for landing zone (p. 1465)
- AWS Control Tower addresses mixed governance for OUs (p. 1465)
- Additional proactive controls available (p. 1465)
- Updated Amazon EC2 proactive controls (p. 1467)
- Seven additional AWS Regions available (p. 1467)
- Account Factory for Terraform (AFT) account customization request tracing (p. 1468)
AWS Control Tower available in Asia Pacific (Melbourne) Region

November 3, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower is available in Asia Pacific (Melbourne) Region.

If you are already using AWS Control Tower and you want to extend its governance features to this Region in your accounts, go to the Settings page in your AWS Control Tower dashboard, select the Region, and then update your landing zone. After a landing zone update, you must update all accounts that are governed by AWS Control Tower, to bring your accounts and OUs under governance in the new Region. For more information, see About Updates.

For a full list of Regions in which AWS Control Tower is available, see the AWS Region Table.

Transition to new AWS Service Catalog External product type

October 31, 2023

(No update required for AWS Control Tower landing zone.)

HashiCorp updated their Terraform licensing. As a result, AWS Service Catalog updated support for Terraform Open Source products and provisioned products to a new product type, called External.

AWS Control Tower does not support Account Factory customizations that rely on the AWS Service Catalog External product type. To avoid disruption to existing workloads and AWS resources in your accounts, follow the AWS Control Tower transition steps in this suggested order, by December 14, 2023:

1. Upgrade your existing Terraform Reference Engine for AWS Service Catalog to include support for both External and Terraform Open Source product types. For instructions about updating your Terraform Reference Engine, review the AWS Service Catalog GitHub Repository.
2. Go to AWS Service Catalog and duplicate any existing Terraform Open Source blueprints to use the new External product type. Do not terminate the existing Terraform Open Source blueprints.
3. Continue to use your existing Terraform Open Source blueprints to create or update accounts in AWS Control Tower.

New control API available

October 14, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports an additional API that you can use to deploy and manage your AWS Control Tower controls, at scale. For more information about the AWS Control Tower control APIs, see the API Reference.

AWS Control Tower added a new control API.
• GetEnabledControl—The API call provides details about an enabled control.

We also updated this API:

ListEnabledControls—This API call lists the controls enabled by AWS Control Tower on the specified organizational unit and the accounts it contains. It now returns additional information in an EnabledControlSummary object.

With these APIs, you can perform several common operations programmatically. For example:

• Get a list of all the controls you’ve enabled from the AWS Control Tower controls library.
• For any enabled control, you can get information about the Regions in which the control is supported, the control's identifier (ARN), the drift status of the control, and the control's status summary.

AWS Control Tower adds additional controls

October 5, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower announces new proactive and detective controls.

Proactive controls in AWS Control Tower are implemented by means of AWS CloudFormation Hooks, which identify and block non-compliant resources before AWS CloudFormation provisions them. Proactive controls complement existing preventive and detective control capabilities in AWS Control Tower.

New proactive controls

• [CT.ATHENA.PR.1] Require an Amazon Athena workgroup to encrypt Athena query results at rest
• [CT.ATHENA.PR.2] Require an Amazon Athena workgroup to encrypt Athena query results at rest with an AWS Key Management Service (KMS) key
• [CT.CLOUDTRAIL.PR.4] Require an AWS CloudTrail Lake event data store to enable encryption at rest with an AWS KMS key
• [CT.DAX.PR.2] Require an Amazon DAX cluster to deploy nodes to at least three Availability Zones
• [CT.EC2.PR.14] Require an Amazon EBS volume configured through an Amazon EC2 launch template to encrypt data at rest
• [CT.EKS.PR.2] Require an Amazon EKS cluster to be configured with secret encryption using AWS Key Management Service (KMS) keys
• [CT.ELASTICLOADBALANCING.PR.14] Require a Network Load Balancer to have cross-zone load balancing activated
• [CT.ELASTICLOADBALANCING.PR.15] Require that an Elastic Load Balancing v2 target group does not explicitly disable cross-zone load balancing
• [CT.EMR.PR.1] Require that an Amazon EMR (EMR) security configuration is configured to encrypt data at rest in Amazon S3
• [CT.EMR.PR.2] Require that an Amazon EMR (EMR) security configuration is configured to encrypt data at rest in Amazon S3 with an AWS KMS key
• [CT.EMR.PR.3] Require that an Amazon EMR (EMR) security configuration is configured with EBS volume local disk encryption using an AWS KMS key
• [CT.EMR.PR.4] Require that an Amazon EMR (EMR) security configuration is configured to encrypt data in transit
• [CT.GLUE.PR.1] Require an AWS Glue job to have an associated security configuration
New drift type reported: trusted access disabled

- **[CT.GLUE.PR.2]** Require an AWS Glue security configuration to encrypt data in Amazon S3 targets using AWS KMS keys
- **[CT.KMS.PR.2]** Require that an AWS KMS asymmetric key with RSA key material used for encryption has a key length greater than 2048 bits
- **[CT.KMS.PR.3]** Require an AWS KMS key policy to have a statement that limits creation of AWS KMS grants to AWS services
- **[CT.LAMBDA.PR.4]** Require an AWS Lambda layer permission to grant access to an AWS organization or specific AWS account
- **[CT.LAMBDA.PR.5]** Require an AWS Lambda function URL to use AWS IAM-based authentication
- **[CT.LAMBDA.PR.6]** Require an AWS Lambda function URL CORS policy to restrict access to specific origins
- **[CT.NEPTUNE.PR.4]** Require an Amazon Neptune DB cluster to enable Amazon CloudWatch log export for audit logs
- **[CT.NEPTUNE.PR.5]** Require an Amazon Neptune DB cluster to set a backup retention period greater than or equal to seven days
- **[CT.REDSHIFT.PR.9]** Require that an Amazon Redshift cluster parameter group is configured to use Secure Sockets Layer (SSL) for encryption of data in transit

These new proactive controls are available in commercial AWS Regions where AWS Control Tower is available. For more details about these controls, see Proactive controls (p. 230). For more details about where the controls are available, see Control limitations (p. 28).

New detective controls

New controls were added to the Security Hub Service-Managed Standard: AWS Control Tower. These controls help you enhance your governance posture. They act as part of the Security Hub Service-Managed Standard: AWS Control Tower, after you enable them on any specific OU.

- **[SH.Athena.1]** Athena workgroups should be encrypted at rest
- **[SH.Neptune.1]** Neptune DB clusters should be encrypted at rest
- **[SH.Neptune.2]** Neptune DB clusters should publish audit logs to CloudWatch Logs
- **[SH.Neptune.3]** Neptune DB cluster snapshots should not be public
- **[SH.Neptune.4]** Neptune DB clusters should have deletion protection enabled
- **[SH.Neptune.5]** Neptune DB clusters should have automated backups enabled
- **[SH.Neptune.6]** Neptune DB cluster snapshots should be encrypted at rest
- **[SH.Neptune.7]** Neptune DB clusters should have IAM database authentication enabled
- **[SH.Neptune.8]** Neptune DB clusters should be configured to copy tags to snapshots
- **[SH.RDS.27]** RDS DB clusters should be encrypted at rest

The new AWS Security Hub detective controls are available in most AWS Regions where AWS Control Tower is available. For more details about these controls, see Controls that apply to Service-Managed Standard: AWS Control Tower. For more details about where the controls are available, see Control limitations (p. 28).

New drift type reported: trusted access disabled

September 21, 2023

(No update required for AWS Control Tower landing zone.)

After you set up your AWS Control Tower landing zone, you can disable trusted access to AWS Control Tower in AWS Organizations. However, doing so causes drift.
With the trusted access disabled drift type, AWS Control Tower notifies you when this type of drift occurs, so you can repair your AWS Control Tower landing zone. For more information, see Types of governance drift.

Four additional AWS Regions

September 13, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower is now available in Asia Pacific (Hyderabad), Europe (Spain and Zurich), and Middle East (UAE).

If you are already using AWS Control Tower and you want to extend its governance features to this Region in your accounts, go to the Settings page in your AWS Control Tower dashboard, select the Region, and then update your landing zone. After a landing zone update, you must update all accounts that are governed by AWS Control Tower to bring your accounts and OUs under governance in the new Region. For more information, see About Updates.

For a full list of Regions in which AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower available in Tel Aviv Region

August 28, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower announces availability in the Israel (Tel Aviv) Region.

If you are already using AWS Control Tower and you want to extend its governance features to this Region in your accounts, go to the Settings page in your AWS Control Tower dashboard, select the Region, and then update your landing zone. After a landing zone update, you must update all accounts that are governed by AWS Control Tower to bring your accounts and OUs under governance in the new Region. For more information, see About Updates.

For a full list of Regions in which AWS Control Tower is available, see the AWS Region Table.

AWS Control Tower launches 28 new proactive controls

July 24, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower is adding 28 new proactive controls, to assist you in managing your AWS environment.

Proactive controls enhance the governance capabilities of AWS Control Tower across your multi-account AWS environments, by blocking non-compliant resources before they are provisioned. These controls help manage AWS services such as Amazon CloudWatch, Amazon Neptune, Amazon ElastiCache, AWS Step Functions, and Amazon DocumentDB. The new controls help you meet control objectives such as establishing logging and monitoring, encrypting data at rest, or improving resiliency.

Here is a full list of the new controls:

- [CT.APPSYNC.PR.1] Require an AWS AppSync GraphQL API to have logging enabled
- [CT.CLOUDWATCH.PR.1] Require an Amazon CloudWatch alarm to have an action configured for the alarm state
AWS Control Tower User Guide

AWS Control Tower launches 28 new proactive controls

- [CT.CLOUDWATCH.PR.2] Require an Amazon CloudWatch log group to be retained for at least one year
- [CT.CLOUDWATCH.PR.3] Require an Amazon CloudWatch log group to be encrypted at rest with an AWS KMS key
- [CT.CLOUDWATCH.PR.4] Require an Amazon CloudWatch alarm action to be activated
- [CT.DOCUMENTDB.PR.1] Require an Amazon DocumentDB cluster to be encrypted at rest
- [CT.DOCUMENTDB.PR.2] Require an Amazon DocumentDB cluster to have automatic backups enabled
- [CT.DYNAMODB.PR.2] Require an Amazon DynamoDB table to be encrypted at rest using AWS KMS keys
- [CT.EC2.PR.13] Require an Amazon EC2 instance to have detailed monitoring enabled
- [CT.EKS.PR.1] Require an Amazon EKS cluster to be configured with public access disabled to the cluster Kubernetes API server endpoint
- [CT.ELASTICACHE.PR.1] Require an Amazon ElastiCache for Redis cluster to have automatic backups activated
- [CT.ELASTICACHE.PR.2] Require an Amazon ElastiCache for Redis cluster to have automatic minor version upgrades activated
- [CT.ELASTICACHE.PR.3] Require an Amazon ElastiCache for Redis replication group to have automatic failover activated
- [CT.ELASTICACHE.PR.4] Require an Amazon ElastiCache replication group to have encryption at rest activated
- [CT.ELASTICACHE.PR.5] Require an Amazon ElastiCache for Redis replication group to have encryption in transit activated
- [CT.ELASTICACHE.PR.6] Require an Amazon ElastiCache cache cluster to use a custom subnet group
- [CT.ELASTICACHE.PR.7] Require an Amazon ElastiCache replication group of earlier Redis versions to have Redis AUTH authentication
- [CT.ELASTICBEANSTALK.PR.3] Require an AWS Elastic Beanstalk environment to have a logging configuration
- [CT.LAMBDA.PR.3] Require an AWS Lambda function to be in a customer-managed Amazon Virtual Private Cloud (VPC)
- [CT.NEPTUNE.PR.1] Require an Amazon Neptune DB cluster to have AWS Identity and Access Management (IAM) database authentication
- [CT.NEPTUNE.PR.2] Require an Amazon Neptune DB cluster to have deletion protection enabled
- [CT.NEPTUNE.PR.3] Require an Amazon Neptune DB cluster to have storage encryption enabled
- [CT.REDSHIFT.PR.8] Require an Amazon Redshift cluster to be encrypted
- [CT.S3.PR.9] Require that an Amazon S3 bucket has S3 Object Lock activated
- [CT.S3.PR.10] Require an Amazon S3 bucket to have server-side encryption configured using AWS KMS keys
- [CT.S3.PR.11] Require an Amazon S3 bucket to have versioning enabled
- [CT.STEPFUNCTIONS.PR.1] Require an AWS Step Functions state machine to have logging activated
- [CT.STEPFUNCTIONS.PR.2] Require an AWS Step Functions state machine to have AWS X-Ray tracing activated

Proactive controls in AWS Control Tower are implemented by means of AWS CloudFormation Hooks, which identify and block non-compliant resources before AWS CloudFormation provisions them. Proactive controls complement existing preventive and detective control capabilities in AWS Control Tower.

These new proactive controls are available in all AWS Regions where AWS Control Tower is available. For more details about these controls, see Proactive controls (p. 230).
AWS Control Tower deprecates two controls

July 18, 2023

AWS Control Tower deprecates two controls

AWS Control Tower conducts regular reviews of its security controls to ensure that they are up to date and are still considered best practices. The following two controls have been deprecated, effective July 18, 2023, and they will be removed from the controls library, effective August 18, 2023. You can no longer enable these controls on any organizational units. You can choose to deactivate these controls before the removal date.

- [SH.S3.4] S3 buckets should have server-side encryption enabled
- [CT.S3.PR.7] Require an Amazon S3 bucket to have server-side encryption configured

Reason for deprecation

As of January 2023, Amazon S3 configured default encryption on all new and existing unencrypted buckets to apply server-side encryption with S3 managed keys (SSE-S3) as the base level of encryption for new objects uploaded to these buckets. No changes have been made to the default encryption configuration for an existing bucket that already had SSE-S3 or server-side encryption with AWS Key Management Service (AWS KMS) keys (SSE-KMS) configured.

AWS Control Tower landing zone version 3.2

June 16, 2023

AWS Control Tower landing zone version 3.2 brings the controls that are part of the AWS Security Hub Service-Managed Standard: AWS Control Tower to general availability. It introduces the ability to view the drift status of controls that are part of this standard in the AWS Control Tower console.

This update includes a new service-linked role (SLR), called the AWSServiceRoleForAWSControlTower. This role assists AWS Control Tower by creating an EventBridge Managed Rule, called the AWSControlTowerManagedRule in each member account. This managed rule collects AWS Security Hub Finding events, from with AWS Control Tower can determine control drift.

This rule is the first managed rule to be created by AWS Control Tower. The rule is not deployed by a stack; it is deployed directly from the EventBridge APIs. You can view the rule in the EventBridge console, or by means of the EventBridge APIs. If the managed-by field is populated, it will show the AWS Control Tower service principal.

Previously, AWS Control Tower assumed the AWSControlTowerExecution role to perform operations in member accounts. This new role and rule are better aligned with the best practices principle of allowing least privilege when performing operations in a multi-account AWS environment. The new role provides scoped-down permissions that specifically allow: creating the managed rule in member accounts, maintaining the managed rule, publishing security notifications through SNS, and verifying drift. For more information, see AWSServiceRoleForAWSControlTower (p. 1403).

The landing zone 3.2 update also includes a new StackSet resource in the management account, BP_BASELINE_SERVICE_LINKED_ROLE, which initially deploys the service-linked role.

When reporting Security Hub control drift (in landing zone 3.2 and later), AWS Control Tower receives a daily status update from Security Hub. Although controls are active in every governed Region, AWS
AWS Control Tower User Guide
AWS Control Tower handles accounts based on ID

Control Tower sends the AWS Security Hub Finding events to the AWS Control Tower home Region only. For more information, see Security Hub control drift reporting (p. 1342).

Update to the Region Deny control
This landing zone version also includes an update to the Region Deny control.

Global services and APIs added
- AWS Billing and Cost Management (billing:*)
- AWS CloudTrail (cloudtrail:LookupEvents) to allow visibility of global events in member accounts.
- AWS Consolidated Billing (consolidatedbilling:*)
- AWS Management Console Mobile Application (consoleapp:*)
- AWS Free Tier (freetier:*)
- AWS Invoicing (invoicing:*)
- AWS IQ (iq:*)
- AWS User Notifications (notifications:*)
- AWS User Notifications Contacts (notifications-contacts:*)
- Amazon Payments (payments:*)
- AWS Tax Settings (tax:*)

Global services and APIs removed
- Removed s3:GetAccountPublic because it is not a valid action.
- Removed s3:PutAccountPublic because it is not a valid action.

AWS Control Tower handles accounts based on ID

June 14, 2023
(No update required for AWS Control Tower landing zone.)

AWS Control Tower now creates and manages accounts that you create in Account Factory by tracking the AWS account ID, rather than the account's email address.

When provisioning an account, the account requester always must have the CreateAccount and the DescribeCreateAccountStatus permissions. This permission set is part of the Admin role, and it is given automatically when a requester assumes the Admin role. If you delegate permission to provision accounts, you may need to add these permissions directly for the account requestors.

Additional Security Hub detective controls available in the AWS Control Tower controls library

June 12, 2023
(No update required for AWS Control Tower landing zone.)

AWS Control Tower has added ten new AWS Security Hub detective controls to the AWS Control Tower controls library. These new controls target services such as API Gateway, AWS CodeBuild, Amazon Elastic Compute Cloud (EC2), Amazon Elastic Load Balancer, Amazon Redshift, Amazon SageMaker, and AWS WAF. These new controls help you enhance your governance posture by meeting control objectives, such as Establish logging and monitoring, Limit network access, and Encrypt data at rest.
These controls act as part of the Security Hub Service-Managed Standard: AWS Control Tower, after you enable them on any specific OU.

- [SH.Account.1] Security contact information should be provided for an AWS account
- [SH.APIGateway.8] API Gateway routes should specify an authorization type
- [SH.APIGateway.9] Access logging should be configured for API Gateway V2 Stages
- [SH.CodeBuild.3] CodeBuild S3 logs should be encrypted
- [SH.EC2.25] EC2 launch templates should not assign public IPs to network interfaces
- [SH.ELB.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS
- [SH.Redshift.10] Redshift clusters should be encrypted at rest
- [SH.SageMaker.2] SageMaker notebook instances should be launched in a custom VPC
- [SH.SageMaker.3] Users should not have root access to SageMaker notebook instances
- [SH.WAF.10] A WAFV2 web ACL should have at least one rule or rule group

The new AWS Security Hub detective controls are available in all AWS Regions where AWS Control Tower is available. For more details about these controls, see Controls that apply to Service-Managed Standard: AWS Control Tower.

AWS Control Tower publishes control metadata tables

June 7, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now provides full tables of control metadata as part of the published documentation. When working with the control APIs, you can look up each control's API controlIdentifier, which is a unique ARN associated with each AWS Region. The tables include the frameworks and control objectives that each control covers. Previously, this information was available in the console only.

The tables also include the metadata for Security Hub controls that are part of the AWS Security Hub Service-Managed Standard:AWS Control Tower. For full details, see Tables of control metadata (p. 1501).

For an abbreviated list of control identifiers, and some usage examples, see Resource identifiers for APIs and controls (p. 200).

Terraform support for Account Factory Customization

June 6, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower offers single-Region support for Terraform through Account Factory Customization (AFC). Starting with this release, you can use AWS Control Tower and Service Catalog together, to define AFC account blueprints, in Terraform open source. You can customize your new and existing AWS accounts, before you provision resources in AWS Control Tower. By default, this feature enables you to deploy and update accounts, with Terraform, in your AWS Control Tower home Region.

An account blueprint describes the specific resources and configurations that are required when an AWS account is provisioned. You can use the blueprint as a template to create multiple AWS accounts at scale.
To get started, use the Terraform Reference Engine on GitHub. The Reference Engine configures the code and infrastructure required for the Terraform open source engine to work with Service Catalog. This one-time setup process takes a few minutes. After that, you can define your custom account requirements in Terraform, and then deploy your accounts with the well-defined AWS Control Tower account factory workflow. Customers who prefer to work with Terraform can utilize AWS Control Tower account customization at scale with AFC, and gain immediate access to each account after it is provisioned.

To learn how to create these customizations, see Creating Products and Getting started with Terraform open source in the Service Catalog documentation. This feature is available in all AWS Regions where AWS Control Tower is available.

AWS IAM Identity Center self-management available for landing zone

June 6, 2023

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports an optional choice of identity provider for an AWS Control Tower landing zone, which you can configure during setup or update. By default, the landing zone is opted-in to using AWS IAM Identity Center, in alignment with best-practices guidance defined in Organizing Your AWS Environment Using Multiple Accounts. You now have three alternatives:

- You can accept the default and allow AWS Control Tower to set up and manage AWS IAM Identity Center for you.
- You can choose to self-manage AWS IAM Identity Center, to reflect your specific business requirements.
- You can optionally bring and self-manage a third-party identity provider, by connecting it through IAM Identity Center, if needed. You should use identity provider optionality if your regulatory environment requires you to use a specific provider, or if you operate in AWS Regions where AWS IAM Identity Center is not available.

For more information, see IAM Identity Center guidance (p. 43).

Selection of identity providers at the account level is not supported. This feature applies only for the landing zone as a whole. AWS Control Tower identity provider optionality is available in all AWS Regions where AWS Control Tower is available.

AWS Control Tower addresses mixed governance for OUs

June 1, 2023

(No update required for AWS Control Tower landing zone.)

With this release, AWS Control Tower prevents controls from deploying to an organizational unit (OU), if that OU is in a state of mixed governance. Mixed governance occurs in an OU if accounts are not updated after AWS Control Tower extends governance to a new AWS Region, or removes governance. This release helps you keep the member accounts of that OU in uniform compliance. For more information, see Avoid mixed governance when configuring Regions (p. 99).

Additional proactive controls available

May 19, 2023
AWS Control Tower User Guide
Additional proactive controls available

(No update required for AWS Control Tower landing zone.)

AWS Control Tower is adding 28 new proactive controls to assist you in governing your multi-account environment and meeting specific control objectives, such as data encryption at rest, or limiting network access. Proactive controls are implemented with AWS CloudFormation hooks that check your resources before they are provisioned. The new controls can help govern AWS services such as Amazon OpenSearch Service, Amazon EC2 Auto Scaling, Amazon SageMaker, Amazon API Gateway, and Amazon Relational Database Service (RDS).

Proactive controls are supported in all commercial AWS Regions where AWS Control Tower is available.

**Amazon OpenSearch Service**

- [CT.OPENSEARCH.PR.1] Require an Elasticsearch domain to encrypt data at rest
- [CT.OPENSEARCH.PR.2] Require an Elasticsearch domain to be created in a user-specified Amazon VPC
- [CT.OPENSEARCH.PR.3] Require an Elasticsearch domain to encrypt data sent between nodes
- [CT.OPENSEARCH.PR.4] Require an Elasticsearch domain to send error logs to Amazon CloudWatch Logs
- [CT.OPENSEARCH.PR.5] Require an Elasticsearch domain to send audit logs to Amazon CloudWatch Logs
- [CT.OPENSEARCH.PR.6] Require an Elasticsearch domain to have zone awareness and at least three data nodes
- [CT.OPENSEARCH.PR.7] Require an Elasticsearch domain to have at least three dedicated master nodes
- [CT.OPENSEARCH.PR.8] Require an Elasticsearch Service domain to use TLSv1.2
- [CT.OPENSEARCH.PR.9] Require an Amazon OpenSearch Service domain to encrypt data at rest
- [CT.OPENSEARCH.PR.10] Require an Amazon OpenSearch Service domain to be created in a user-specified Amazon VPC
- [CT.OPENSEARCH.PR.11] Require an Amazon OpenSearch Service domain to encrypt data sent between nodes
- [CT.OPENSEARCH.PR.12] Require an Amazon OpenSearch Service domain to send error logs to Amazon CloudWatch Logs
- [CT.OPENSEARCH.PR.13] Require an Amazon OpenSearch Service domain to send audit logs to Amazon CloudWatch Logs
- [CT.OPENSEARCH.PR.14] Require an Amazon OpenSearch Service domain to have zone awareness and at least three data nodes
- [CT.OPENSEARCH.PR.15] Require an Amazon OpenSearch Service domain to use fine-grained access control
- [CT.OPENSEARCH.PR.16] Require an Amazon OpenSearch Service domain to use TLSv1.2

**Amazon EC2 Auto Scaling**

- [CT.AUTOISCALING.PR.1] Require an Amazon EC2 Auto Scaling group to have multiple Availability Zones
- [CT.AUTOISCALING.PR.2] Require an Amazon EC2 Auto Scaling group launch configuration to configure Amazon EC2 instances for IMDSv2
- [CT.AUTOISCALING.PR.3] Require an Amazon EC2 Auto Scaling launch configuration to have a single-hop metadata response limit
- [CT.AUTOISCALING.PR.4] Require an Amazon EC2 Auto Scaling group associated with an Amazon Elastic Load Balancing (ELB) to have ELB health checks activated
- [CT.AUTOISCALING.PR.5] Require that an Amazon EC2 Auto Scaling group launch configuration does not have Amazon EC2 instances with public IP addresses
- [CT.AUTOISCALING.PR.6] Require any Amazon EC2 Auto Scaling groups to use multiple instance types
• [CT.AUTOSCALING.PR.8] Require an Amazon EC2 Auto Scaling group to have EC2 launch templates configured

**Amazon SageMaker**

• [CT.SAGEMAKER.PR.1] Require an Amazon SageMaker notebook instance to prevent direct internet access
• [CT.SAGEMAKER.PR.2] Require Amazon SageMaker notebook instances to be deployed within a custom Amazon VPC
• [CT.SAGEMAKER.PR.3] Require Amazon SageMaker notebook instances to have root access disallowed

**Amazon API Gateway**

• [CT.APIGATEWAY.PR.5] Require Amazon API Gateway V2 Websocket and HTTP routes to specify an authorization type

**Amazon Relational Database Service (RDS)**

• [CT.RDS.PR.25] Require an Amazon RDS database cluster to have logging configured

For more information, see Proactive controls (p. 230).

### Updated Amazon EC2 proactive controls

**May 2, 2023**

(No update required for AWS Control Tower landing zone.)

AWS Control Tower has updated two proactive controls: CT.EC2.PR.3 and CT.EC2.PR.4.

For the updated CT.EC2.PR.3 control, any AWS CloudFormation deployment that references a prefix list for a security group resource is blocked from deployment, unless it is for port 80 or 443.

For the updated CT.EC2.PR.4 control, any AWS CloudFormation deployment that references a prefix list for a security group resource is blocked if the port is 3389, 20, 23, 110, 143, 3306, 8080, 1433, 9200, 9300, 25, 445, 135, 21, 1434, 4333, 5432, 5500, 5601, 22, 3000, 5000, 8088, 8888.

### Seven additional AWS Regions available

**April 19, 2023**

(No update required for AWS Control Tower landing zone.)

AWS Control Tower is now available in seven additional AWS Regions: Northern California (San Francisco), Asia Pacific (Hong Kong, Jakarta, and Osaka), Europe (Milan), Middle East (Bahrain), and Africa (Cape Town). These additional Regions for AWS Control Tower, called opt-in Regions, are not active by default, except the US West (N. California) Region, which is active by default.

Some controls in AWS Control Tower do not operate in some of these additional AWS Regions where AWS Control Tower is available, because those Regions do not support the required underlying functionality. For details, see Control limitations (p. 28).

Among these new Regions, CfCT is not available in Asia Pacific (Jakarta and Osaka). Availability in other AWS Regions is unchanged.
For more information about how AWS Control Tower manages the limitations of Regions and controls, see [Considerations for activating AWS opt-in Regions (p. 100)](#).

The VPCe endpoints required by AFT are not available in the Middle East (Bahrain) Region. Customers deploying AFT in this Region are required to deploy with parameter `aft_vpc_endpoints=false`. For more information, see the parameter in [the README file](#).

AWS Control Tower VPCs have two Availability Zones in the US West (N. California) Region, `us-west-1`, due to a limitation in Amazon EC2. In the US West (N. California), six subnets are divided across two Availability Zones. For more information, see [Overview of AWS Control Tower and VPCs (p. 84)](#).

AWS Control Tower added new permissions to `AWSControlTowerServiceRolePolicy` that allow AWS Control Tower to make calls to the `EnableRegion`, `ListRegions`, and `GetRegionOptStatus` APIs implemented by the AWS Account Management service, to make these additional AWS Regions available for your shared accounts in the landing zone (Management account, Log archive account, Audit account) and your OU member accounts. For more information, see [Managed policies for AWS Control Tower](#).

## Account Factory for Terraform (AFT) account customization request tracing

February 16, 2023

AFT supports account customization request tracing. Every time you submit an account customization request, AFT generates a unique tracing token that passes through an AFT customization AWS Step Functions state machine, which logs the token as part of its execution. You can use Amazon CloudWatch Logs insights queries to search timestamp ranges and retrieve the request token. As a result, you can see payloads that accompany the token, so you can trace your account customization request throughout the entire AFT workflow. For more information about AFT, see [Overview of AWS Control Tower Account Factory for Terraform](#). For information about CloudWatch Logs and Step Functions, see the following:

- [What is Amazon CloudWatch Logs?](#) in the [Amazon CloudWatch Logs User Guide](#)
- [What is AWS Step Functions?](#) in the [AWS Step Functions Developer Guide](#)

## AWS Control Tower landing zone version 3.1

February 9, 2023

(Update required for AWS Control Tower landing zone to version 3.1. For information, see [Update Your Landing Zone (p. 47)](#))

AWS Control Tower landing zone version 3.1 includes the following updates:

- With this release, AWS Control Tower deactivates unnecessary access logging for your `access logging bucket`, which is the Amazon S3 bucket where access logs are stored in the Log Archive account, while continuing to enable server access logging for S3 buckets. This release also includes updates to the Region Deny control that allow additional actions for global services, such as AWS Support Plans and AWS Artifact.
- Deactivation of server access logging for the AWS Control Tower access logging bucket causes Security Hub to create a finding for the Log Archive account's `access logging bucket`, due to an AWS Security Hub rule, [S3.9] S3 bucket server access logging should be enabled. In alignment with Security Hub, we recommend that you suppress this particular finding, as stated in the Security Hub description of this rule. For additional information, see [information about suppressed findings](#).
- Access logging for the (regular) logging bucket in the Log Archive account is unchanged in version 3.1. In alignment with best practices, access events for that bucket are recorded as log entries in the `access`
Proactive controls generally available

January 24, 2023

(No update required for AWS Control Tower landing zone.)

Optional proactive controls, previously announced in preview status, are now generally available. These controls are referred to as proactive because they check your resources – before the resources are deployed – to determine whether the new resources comply with the controls that are activated in your environment. For more information, see Comprehensive controls assist in AWS resource provisioning and management (p. 1471).

January - December 2022

In 2022, AWS Control Tower released the following updates:

- Concurrent account operations (p. 1470)
- Account Factory Customization (AFC) (p. 1470)
- Comprehensive controls assist in AWS resource provisioning and management (p. 1471)
Concurrent account operations

December 16, 2022

AWS Control Tower now supports concurrent actions in account factory. You can create, update, or enroll up to five (5) accounts at a time. Submit up to five actions in succession and view the completion status of each request, while your accounts finish building in the background. For example, you no longer must wait for each process to complete before you update another account, or before you re-register an entire organizational unit (OU).

Account Factory Customization (AFC)

November 28, 2022

Account factory customization allows you to customize new and existing accounts from within the AWS Control Tower console. These new customization capabilities give you the flexibility to define account blueprints, which are AWS CloudFormation templates contained in a specialized Service Catalog product. Blueprints provision fully customized resources and configurations. You also may choose use pre-defined blueprints, built and managed by AWS partners, that help you customize accounts for specific use cases.

Previously, AWS Control Tower account factory did not support account customization in the console. With this update of account factory, you can pre-define account requirements and implement them as part of a well-defined workflow. You can apply blueprints to create new accounts, to enroll other AWS accounts into AWS Control Tower, and to update existing AWS Control Tower accounts.

When you provision, enroll, or update an account in account factory, you will select the blueprint to deploy. Those resources specified in the blueprint are provisioned in your account. When your account has finished building, all of the custom configurations are available for use immediately.

To get started with customizing accounts, you can define the resources for your intended use case in a Service Catalog product. You also can select partner-managed solutions from the AWS Getting Started Library. For more information, see Customize accounts with Account Factory Customization (AFC) (p. 129).
Comprehensive controls assist in AWS resource provisioning and management

November 28, 2022

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports comprehensive controls management, including new, optional proactive controls, implemented through AWS CloudFormation hooks. These controls are referred to as proactive because they check your resources – before the resources are deployed – to determine whether the new resources will comply with the controls that are activated in your environment.

Over 130 new proactive controls assist you with meeting specific policy objectives for your AWS Control Tower environment; with meeting requirements of industry-standard compliance frameworks; and with governing AWS Control Tower interactions across more than twenty other AWS services.

The AWS Control Tower controls library classifies these controls according to the associated AWS services and resources. For more details, see Proactive controls (p. 230).

With this release, AWS Control Tower also is integrated with AWS Security Hub, by means of the new Security Hub Service-Managed Standard: AWS Control Tower, which supports the AWS Foundational Security Best Practices (FSBP) standard. You can view over 160 Security Hub controls alongside AWS Control Tower controls in the console, and you can obtain a Security Hub security score for your AWS Control Tower environment. For more information, see Security Hub standard (p. 1341).

Compliance status viewable for all AWS Config rules

November 18, 2022

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now displays the compliance status of all AWS Config rules deployed into organizational units registered with AWS Control Tower. You can view the compliance status of all AWS Config rules that affect your accounts in AWS Control Tower, enrolled or unenrolled, without navigating outside of the AWS Control Tower console. Customers can choose to set up Config rules, called detective controls, in AWS Control Tower, or to set them up directly through the AWS Config service. The rules deployed by AWS Config are shown, along with the rules deployed by AWS Control Tower.

Previously, AWS Config rules deployed through the AWS Config service were not visible in the AWS Control Tower console. Customers had to navigate to the AWS Config service to identify non-compliant AWS Config rules. Now you can identify any non-compliant AWS Config rule within the AWS Control Tower console. To view the compliance status of all your Config rules, navigate to the Account details page in the AWS Control Tower console. You will see a list showing the compliance status of controls managed by AWS Control Tower and Config rules deployed outside of AWS Control Tower.

API for controls and a new AWS CloudFormation resource

September 1, 2022

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now supports programmatic management of controls, also known as guardrails, through a set of API calls. A new AWS CloudFormation resource supports the API functionality for controls. For more details, see Automate tasks in AWS Control Tower (p. 50) and Creating AWS Control Tower resources with AWS CloudFormation (p. 54).
These APIs allow you to enable, disable, and view the application status of controls in the AWS Control Tower library. The APIs include support for AWS CloudFormation, so you can manage AWS resources as infrastructure-as-code (IaC). AWS Control Tower provides optional preventive and detective controls that express your policy intentions regarding an entire organizational unit (OU), and every AWS account within the OU. These rules remain in effect as you create new accounts or make changes to existing accounts.

APIs included in this release

- **EnableControl**– This API call activates a control. It starts an asynchronous operation that creates AWS resources on the specified organizational unit and the accounts it contains.
- **DisableControl**– This API call turns off a control. It starts an asynchronous operation that deletes AWS resources on the specified organizational unit and the accounts it contains.
- **GetControlOperation**– Returns the status of a particular EnableControl or DisableControl operation.
- **ListEnabledControls**– Lists the controls enabled by AWS Control Tower on the specified organizational unit and the accounts it contains.

To view a list of control names for optional controls, see Resource identifiers for APIs and controls (p. 200), in the AWS Control Tower User Guide.

CfCT supports stack set deletion

August 26, 2022

(No update required for AWS Control Tower landing zone.)

Customizations for AWS Control Tower (CfCT) now supports stack set deletion, by setting a parameter in the manifests.yaml file. For more information, see Delete a stack set (p. 67).

**Important**

When you initially set the value of enable_stack_set_deletion to true, the next time you invoke CfCT, **ALL** resources that begin with the prefix CustomControlTower-, which have the associated key tag Key:AWS_Solutions, Value: CustomControlTowerStackSet, and which are not declared in the manifest file, are staged for deletion.

Customized log retention

August 15, 2022

(Update required for AWS Control Tower landing zone. For information, see Update Your Landing Zone (p. 47))

AWS Control Tower now provides the ability to customize the retention policy for Amazon S3 buckets that store your AWS Control Tower CloudTrail logs. You can customize your Amazon S3 log retention policy, in increments of days or years, up to a maximum of 15 years.

If you choose not customize your log retention, the default settings are 1 year for standard account logging, and 10 years for access logging.

This feature is available for existing customers through AWS Control Tower when you update or repair your landing zone, and for new customers through the AWS Control Tower setup process.

Role drift repair available

August 11, 2022
AWS Control Tower now supports repair for role drift. You can restore a required role without a full repair of your landing zone. If this type of drift repair is needed, the console error page provides steps for restoring the role, so that your landing zone is once again available.

AWS Control Tower landing zone version 3.0

July 29, 2022

AWS Control Tower landing zone version 3.0 includes the following updates:

- The option to choose organization-level AWS CloudTrail trails, or to opt out of CloudTrail trails managed by AWS Control Tower.
- Two new detective controls to determine whether AWS CloudTrail is logging activity in your accounts.
- The option to aggregate AWS Config information about global resources in your home Region only.
- An update to the Region deny control.
- An update to the managed policy, `AWSControlTowerServiceRolePolicy`.
- We no longer create the IAM role `aws-controltower-CloudWatchLogsRole` and the CloudWatch log group `aws-controltower/CloudTrailLogs` in each enrolled account. Previously, we created these in each account for its account trail. With organization trails, we only create one in the management account.

The following sections provide more details about each new capability.

Organization-level CloudTrail trails in AWS Control Tower

With landing zone version 3.0, AWS Control Tower now supports organization-level AWS CloudTrail trails.

When you update your AWS Control Tower landing zone to version 3.0, you have the option to select organization-level AWS CloudTrail trails as your logging preference, or to opt out of CloudTrail trails that are managed by AWS Control Tower. When you update to version 3.0, AWS Control Tower deletes the existing account-level trails for enrolled accounts after a 24-hour waiting period. AWS Control Tower does not delete account-level trails for unenrolled accounts. Going forward from landing zone 3.0, AWS Control Tower no longer will support account-level trails that AWS manages. Instead, AWS Control Tower creates an organization-level trail, which is active or inactive, according to your selection.

Note

After you update to version 3.0 or later, you do not have the option to continue with account-level CloudTrail trails managed by AWS Control Tower.

No logging data is lost from your aggregated account logs, because the logs remain in the existing Amazon S3 bucket where they are stored. Only the trails are deleted, not the existing logs. If you select the option to add organization-level trails, AWS Control Tower opens a new path to a new folder within your Amazon S3 bucket and continues sending logging information to that location. If you choose to opt out of trails managed by AWS Control Tower, your existing logs remain in the bucket, unchanged.

Path naming conventions for log storage

- Account trail logs are stored with a path of this form: `/*org id*/AWSLogs/…`
- Organization trail logs are stored with a path of this form: `/*org id*/AWSLogs/*org id*/…`
The path that AWS Control Tower creates for your organization-level CloudTrail trails is different than the default path for a manually-created organization-level trail, which would have the following form:

- /AWSLogs/org_id/...

For more information about CloudTrail path naming, see Finding your CloudTrail log files.

Tip
If you plan to create and manage your own account-level trails, we recommend that you create the new trails before you complete the update to AWS Control Tower landing zone version 3.0, to start logging right away.

At any time, you may choose to create new account-level or organization-level CloudTrail trails and manage them on your own. The option to choose organization-level CloudTrail trails managed by AWS Control Tower is available during any landing zone update to version 3.0 or later. You can opt into and opt out of organization-level trails, whenever you update your landing zone.

If your logs are managed by a third-party service, be sure to give the new path name to your service.

Note
For landing zones at version 3.0 or later, account-level AWS CloudTrail trails are not supported by AWS Control Tower. You can create and maintain your own account-level trails at any time, or you can opt into organization-level trails managed by AWS Control Tower.

Record AWS Config resources in the home Region only

In landing zone version 3.0, AWS Control Tower has updated the baseline configuration for AWS Config so that it records global resources in the home Region only. After you update to version 3.0, resource recording for global resources is enabled only in your home Region.

This configuration is considered a best practice. It is recommended by AWS Security Hub and AWS Config, and it creates cost savings by reducing the number of configuration items created when global resources are created, modified, or deleted. Previously, each time a global resource was created, updated, or deleted, whether by a customer or by an AWS service, a configuration item was created for each item in each governed Region.

Two new detective controls for AWS CloudTrail logging

As part of the change to organization-level AWS CloudTrail trails, AWS Control Tower is introducing two new detective controls that check whether CloudTrail is enabled. The first control has Mandatory guidance, and it is enabled on the Security OU during setup or landing zone updates of 3.0 and later. The second control has Strongly recommended guidance, and it is optionally applied to any OUs other than the Security OU, which already has the mandatory control protection enforced.

Mandatory control: Detect whether shared accounts under the Security organizational unit have AWS CloudTrail or CloudTrail Lake enabled (p. 229)

Strongly recommended control: Detect whether an account has AWS CloudTrail or CloudTrail Lake enabled (p. 1369)

For more information about the new controls, see the The AWS Control Tower controls library (p. 215).

An update to the Region deny control

We updated the NotAction list in the Region deny control to include actions by some additional services, listed below:

```
"chatbot:**",
"s3:GetAccountPublic",
```
| "s3:DeleteMultiRegionAccessPoint", |
| "s3:DescribeMultiRegionAccessPointOperation", |
| "s3:GetMultiRegionAccessPoint", |
| "s3:GetMultiRegionAccessPointPolicy", |
| "s3:GetMultiRegionAccessPointPolicyStatus", |
| "s3:ListMultiRegionAccessPoints", |
| "s3:GetStorageLensConfiguration", |
| "s3:GetStorageLensDashboard", |
| "s3:ListStorageLensConfigurations" |
| "s3:GetAccountPublicAccessBlock", |
| "s3:PutAccountPublic", |
| "s3:PutAccountPublicAccessBlock" |

### Video Walkthrough

This video (3:07) describes how to update your existing AWS Control Tower landing zone to version 3. For better viewing, select the icon at the lower right corner of the video to enlarge it to full screen. Captioning is available.

**Video Walkthrough of Update an Existing AWS Control Tower Landing Zone to Landing Zone 3.**

### The Organization page combines views of OUs and accounts

**July 18, 2022**

(No update required for AWS Control Tower landing zone)

The new **Organization** page in AWS Control Tower shows a hierarchical view of all organizational units (OUs) and accounts. It combines the information from the **OUs** and **Accounts** pages, which existed previously.

On the new page, you can see relationships between parent OUs and their nested OUs and accounts. You can take action on groupings of resources. You can configure the page view. For example, you can expand or collapse the hierarchical view, filter the view to see accounts or OUs only, choose to view only your enrolled accounts and registered OUs, or you can view groups of related resources. It is easier to ensure that your entire organization is updated properly.

### Easier enroll and update for individual member accounts

**May 31, 2022**

(No update required for AWS Control Tower landing zone)

AWS Control Tower now gives you an improved capability to update and enroll member accounts individually. Each account shows when it is available for an update, so you can more easily ensure that your member accounts include the latest configuration. You can update your landing zone, remediate account drift, or enroll an account into a registered OU, in a few streamlined steps.

When you update an account, there’s no need to include an account’s entire organizational unit (OU) in each update action. As a result, the time required to update an individual account is greatly reduced.

You can enroll accounts into AWS Control Tower OUs with more help from the AWS Control Tower console. Existing accounts that you enroll in AWS Control Tower must still meet the account
prerequisites, and you must add the AWSControlTowerExecution role. Then, you can choose any registered OU and enroll the account into it by selecting the Enroll button.

We’ve separated the Enroll account functionality from the Create account workflow in account factory, to create more distinction between these similar processes, and help avoid setup errors when you’re entering account information.

AFT supports automated customization for shared AWS Control Tower accounts

May 27, 2022

(No update required for AWS Control Tower landing zone)

Account Factory for Terraform (AFT) now can programmatically customize and update any of your accounts that are managed by AWS Control Tower, including the management account, audit account, and log archive account, along with your enrolled accounts. You can centralize your account customization and update management, while protecting the security of your account configurations, because you scope the role that carries out the work.

The existing AWSAFTExecution role now deploys customizations in all accounts. You can set up IAM permissions with boundaries that limit the access of the AWSAFTExecution role according to your business and security requirements. You also can programmatically delegate the approved customization permissions in that role, for trusted users. As a best practice, we recommend that you restrict permissions to those that are necessary to deploy the required customizations.

AFT now creates the new AWSAFTService role to deploy AFT resources in all managed accounts, including the shared accounts and management account. Resources formerly were deployed by the AWSAFTExecution role.

The AWS Control Tower shared and management accounts are not provisioned through account factory, so they do not have corresponding provisioned products in AWS Service Catalog. Therefore, you are not able to update the shared and management accounts in Service Catalog.

Concurrent operations for all optional controls

May 18, 2022

(No update required for AWS Control Tower landing zone)

AWS Control Tower now supports concurrent operations for preventive controls, as well as for detective controls.

With this new feature, any optional control now can be applied or removed concurrently, thereby improving the ease of use and performance for all optional controls. You can enable multiple optional controls without waiting for individual control operations to complete. The only restricted times are when AWS Control Tower is in the process of landing zone setup, or while extending governance to a new organization.

Supported functionality for preventive controls:

• Apply and remove different preventive controls on the same OU.
• Apply and remove different preventive controls on different OUs, concurrently.
• Apply and remove the same preventive control on multiple OUs, concurrently.
• You can apply and remove any preventive and detective controls, concurrently.
You can experience these control concurrency improvements in all released versions of AWS Control Tower.

When you apply preventive controls to nested OUs, the preventive controls affect all accounts and OUs nested under the target OU, even if those accounts and OUs are not registered with AWS Control Tower. Preventive controls are implemented using Service Control Policies (SCPs), which are part of AWS Organizations. Detective controls are implemented using AWS Config rules. Guardrails remain in effect as you create new accounts or make changes to your existing accounts, and AWS Control Tower provides a summary report of how each account conforms to your enabled policies. For a full list of available controls, see the The AWS Control Tower controls library (p. 215).

Existing security and logging accounts

May 16, 2022

(Available during initial setup.)

AWS Control Tower now provides the option for you to specify an existing AWS account as an AWS Control Tower security or logging account, during the initial landing zone setup process. This option eliminates the need for AWS Control Tower to create new, shared accounts. The security account, which is called the Audit account by default, is a restricted account that gives your security and compliance teams access to all accounts in your landing zone. The logging account, which is called the Log Archive account by default, works as a repository. It stores logs of API activities and resource configurations from all accounts in your landing zone.

By bringing your existing security and logging accounts, it is easier to extend AWS Control Tower governance into your existing organizations, or to move to AWS Control Tower from an alternate landing zone. The option for you to use existing accounts is displayed during the initial landing zone setup. It includes checks during the setup process, which ensure successful deployment. AWS Control Tower implements the necessary roles and controls on your existing accounts. It does not remove or merge any existing resources or data that exists in these accounts.

Limitation: If you plan to bring existing AWS accounts into AWS Control Tower as audit and log archive accounts, and if those accounts have existing AWS Config resources, you must delete the existing AWS Config resources before you can enroll the accounts into AWS Control Tower.

AWS Control Tower landing zone version 2.9

April 22, 2022

(Update required for AWS Control Tower landing zone to version 2.9. For information, see Update Your Landing Zone (p. 47))

AWS Control Tower landing zone version 2.9 updates the notification forwarder Lambda to use the Python version 3.9 runtime. This update addresses the deprecation of Python version 3.6, which is planned for July of 2022. For the latest information, see the Python deprecation page.

AWS Control Tower landing zone version 2.8

February 10, 2022

(Update required for AWS Control Tower landing zone to version 2.8. For information, see Update Your Landing Zone (p. 47))

AWS Control Tower landing zone version 2.8 adds functionality that aligns with recent updates to the AWS Foundational Security Best Practices.
In this release:

- Access logging is configured for the access log bucket in the Log Archive account, to keep track of access to the existing S3 access log bucket.
- Support for lifecycle policy is added. The access log for the existing S3 access log bucket is set to a default retention time of 10 years.
- Additionally, this release updates AWS Control Tower to use the AWS Service Linked Role (SLR) provided by AWS Config, in all managed accounts (not including the management account), so that you can set up and manage Config rules to match AWS Config best practices. Customers who do not upgrade will continue to use their existing role.
- This release streamlines the AWS Control Tower KMS configuration process for encrypting Config data, and it improves the related status messaging in CloudTrail.
- The release includes an update to the Region deny control, to allow for the route53-application-recovery feature in us-west-2.
- Update: On February 15, 2022, we removed the dead letter queue for AWS Lambda functions.

Additional details:

- If you decommission your landing zone, AWS Control Tower does not remove the AWS Config service-linked role.
- If you deprovision an Account Factory account, AWS Control Tower does not remove the AWS Config service-linked role.

To update your landing zone to 2.8, navigate to the Landing zone settings page, select the 2.8 version, and then choose Update. After you update your landing zone, you must update all accounts that are governed by AWS Control Tower, as given in Configuration update management in AWS Control Tower (p. 46).

January - December 2021

In 2021, AWS Control Tower released the following updates:

- Region deny capabilities (p. 1479)
- Data residency features (p. 1479)
- AWS Control Tower introduces Terraform account provisioning and customization (p. 1479)
- New lifecycle event available (p. 1480)
- AWS Control Tower enables nested OUs (p. 1480)
- Detective control concurrency (p. 1480)
- Two new Regions available (p. 1481)
- Region deselection (p. 1481)
- AWS Control Tower works with AWS Key Management Systems (p. 1482)
- Controls renamed, functionality unchanged (p. 1482)
- AWS Control Tower scans SCPs daily to check for drift (p. 1482)
- Customized names for OUs and accounts (p. 1483)
- AWS Control Tower landing zone version 2.7 (p. 1483)
- Three new AWS Regions available (p. 1484)
- Govern selected Regions only (p. 1484)
AWS Control Tower User Guide
Region deny capabilities

- AWS Control Tower now extends governance to existing OUs in your AWS organizations (p. 1485)
- AWS Control Tower provides bulk account updates (p. 1485)

Region deny capabilities

November 30, 2021

(No update required for AWS Control Tower landing zone.)

AWS Control Tower now provides Region deny capabilities, which assist you in limiting access to AWS services and operations for enrolled accounts in your AWS Control Tower environment. The Region deny feature complements existing Region selection and Region deselection features in AWS Control Tower. Together, these features help you to address compliance and regulatory concerns, while balancing the costs associated with expanding into additional Regions.

For example, AWS customers in Germany can deny access to AWS services in Regions outside of the Frankfurt Region. You can select restricted Regions during the AWS Control Tower set up process, or in the Landing zone settings page. The Region deny feature is available when you update your AWS Control Tower landing zone version. Select AWS services are exempt from Region deny capabilities. To learn more, see Configure the Region deny control (p. 102).

Data residency features

November 30, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower now offers purpose-built controls to help ensure that any customer data you upload to AWS services is located only in the AWS Regions that you specify. You can select the AWS Region or Regions in which your customer data is stored and processed. For a full list of AWS Regions where AWS Control Tower is available, see the AWS Region Table.

For granular control, you can apply additional controls, such as Disallow Amazon Virtual Private Network (VPN) connections, or Disallow internet access for an Amazon VPC instance. You can view the compliance status of the controls in the AWS Control Tower console. For a full list of available controls, see The AWS Control Tower controls library (p. 215).

AWS Control Tower introduces Terraform account provisioning and customization

November 29, 2021

(Optional update for AWS Control Tower landing zone)

You can now employ Terraform to provision and update customized accounts through AWS Control Tower, with AWS Control Tower Account Factory for Terraform (AFT).

AFT provides a single Terraform infrastructure as code (IaC) pipeline, which provisions accounts managed by AWS Control Tower. Customizations during provisioning help to meet your business and security policies, before you give the accounts to end-users.

The AFT automated account creation pipeline monitors until account provisioning is complete, and then it continues, triggering additional Terraform modules that enhance the account with any necessary customizations. As an additional part of the customization process, you can configure the pipeline to
install your own custom Terraform modules, and you can choose to add any of the AFT Feature Options, which are provided by AWS for common customizations.

Get started with AWS Control Tower Account Factory for Terraform by following the steps provided in the AWS Control Tower User Guide, Deploy AWS Control Tower Account Factory for Terraform (AFT) (p. 142), and by downloading AFT for your Terraform instance. AFT supports Terraform Cloud, Terraform Enterprise, and Terraform Open Source distributions.

New lifecycle event available

November 18, 2021

(No update required for AWS Control Tower landing zone)

The PrecheckOrganizationalUnit event logs whether any resources block the Extend governance task from success, including resources in nested OUs. For more information, see PrecheckOrganizationalUnit (p. 1425).

AWS Control Tower enables nested OUs

November 16, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower now enables you to include nested OUs as part of your landing zone.

AWS Control Tower provides support for nested organizational units (OUs), allowing you to organize accounts into multiple hierarchy levels, and to enforce preventive controls hierarchically. You can register OUs containing nested OUs, create and register OUs under parent OUs, and enable controls on any registered OU, regardless of depth. To support this functionality, the console shows the number of governed accounts and OUs.

With nested OUs, you can align your AWS Control Tower OUs to the AWS multi-account strategy, and you can reduce the time required to enable controls on multiple OUs, by enforcing controls at the parent OU level.

Key considerations

1. You can register existing, multi-level OUs with AWS Control Tower one OU at a time, starting with the top-level OU and then proceeding down the tree. For more information, see Expand from flat OU structure to nested OU structure (p. 186).
2. Accounts directly under a registered OU are enrolled automatically. Accounts further down the tree can be enrolled by registering their immediate parent OU.
3. Preventive controls (SCPs) are inherited down the hierarchy automatically; SCPs applied to the parent are inherited by all nested OUs.
4. Detective controls (AWS Config rules) are NOT inherited automatically.
5. Compliance with detective controls is reported by each OU.
6. SCP drift on an OU affects all accounts and OUs under it.
7. You cannot create new nested OUs under the Security OU (Core OU).

Detective control concurrency

November 5, 2021

(Optional update for AWS Control Tower landing zone)
AWS Control Tower detective controls now support concurrent operations for detective controls, improving the ease of use and performance. You can enable multiple detective controls without waiting for individual control operations to complete.

**Supported functionality:**

- Enable different detective controls on the same OU (for example, Detect Whether MFA for the Root User is Enabled and Detect Whether Public Write Access to Amazon S3 Buckets is Allowed).
- Enable different detective controls on different OUs, concurrently.
- Guardrail error messaging has been improved to give additional guidance for supported control concurrency operations.

**Not supported in this release:**

- Enabling the same detective control on multiple OUs concurrently is not supported.
- Preventive control concurrency is not supported.

You can experience the detective control concurrency improvements in all versions of AWS Control Tower. It is recommended that customers not currently on version 2.7 perform a landing zone update to take advantage of other features, such as Region selection and deselection, which are available in the latest version.

**Two new Regions available**

**July 29, 2021**

(Update required for AWS Control Tower landing zone)

AWS Control Tower is now available in two additional AWS Regions: South America (Sao Paulo), and Europe (Paris). This update expands AWS Control Tower availability to 15 AWS Regions.

If you are new to AWS Control Tower, you can launch it right away in any of the supported Regions. During the launch, you can select the Regions in which you want AWS Control Tower to build and govern your multi-account environment.

If you already have an AWS Control Tower environment and you want to extend or remove AWS Control Tower governance features in one or more supported Regions, go to the **Landing Zone Settings** page in your AWS Control Tower dashboard, then select the Regions. After updating your landing zone, you must then update all accounts that are governed by AWS Control Tower.

**Region deselection**

**July 29, 2021**

(Optional update for AWS Control Tower landing zone)

AWS Control Tower Region deselection enhances your ability to manage the geographical footprint of your AWS Control Tower resources. You can deselect Regions you would no longer like AWS Control Tower to govern. This feature provides you with the capability to address compliance and regulatory concerns while balancing the costs associated with expanding into additional Regions.

Region deselection is available when you update your AWS Control Tower landing zone version.

When you use Account Factory to create a new account or enroll a pre-existing member account, or when you select **Extend Governance** to enroll accounts in a pre-existing organizational unit, AWS Control
AWS Control Tower works with AWS Key Management Systems

July 28, 2021

(Optional update for AWS Control Tower landing zone)

AWS Control Tower provides you the option to use an AWS Key Management Service (AWS KMS) key. A key is provided and managed by you, to secure the services that AWS Control Tower deploys, including AWS CloudTrail, AWS Config, and the associated Amazon S3 data. AWS KMS encryption is an enhanced level of encryption over the SSE-S3 encryption that AWS Control Tower uses by default.

The integration of AWS KMS support into AWS Control Tower aligns with the AWS Foundational Security Best Practices, which recommend an added layer of security for your sensitive log files. You should use AWS KMS–managed keys (SSE-KMS) for encryption at rest. AWS KMS encryption support is available when you set up a new landing zone or when you update your existing AWS Control Tower landing zone.

To configure this functionality, you can select KMS Key Configuration during your initial landing zone setup. You can choose an existing KMS key, or you can select a button that directs you to the AWS KMS console to create a new one. You also have the flexibility to change from default encryption to SSE-KMS, or to a different SSE-KMS key.

For an existing AWS Control Tower landing zone, you can perform an update to start using AWS KMS keys.

Controls renamed, functionality unchanged

July 26, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower is revising certain control names and descriptions to better reflect the policy intentions of the control. The revised names and descriptions help you understand more intuitively the ways in which controls embody the policies of your accounts. For example, we changed part of the names of detective controls from “Disallow” to “Detect” because the detective control itself does not stop a specific action, it only detects policy violations and provides alerts through the dashboard.

Control functionality, guidance, and implementation remain unchanged. Only the control names and descriptions have been revised.

AWS Control Tower scans SCPs daily to check for drift

May 11, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower now performs daily automated scans of your managed SCPs to verify that the corresponding controls are applied correctly and that they have not drifted. If a scan discovers drift, you will receive a notification. AWS Control Tower sends only one notification per drift issue, so if your
AWS Control Tower User Guide

Customized names for OUs and accounts

April 16, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower now allows you to customize your landing zone naming. You can retain the names that AWS Control Tower recommends for the organizational units (OUs) and core accounts, or you can modify these names during the initial landing zone setup process.

The default names that AWS Control Tower provides for the OUs and core accounts match the AWS multi-account best practices guidance. However, if your company has specific naming policies, or if you already have an existing OU or account with the same recommended name, the new OU and account naming functionality gives you the flexibility to address those constraints.

Separately from that workflow change during setup, the OU formerly known as the Core OU is now called the Security OU, and the OU formerly known as the Custom OU is now called the Sandbox OU. We made this change to improve our alignment with overall AWS best practices guidance for naming.

New customers will see these new OU names. Existing customers will continue to see the original names of these OUs. You may encounter some inconsistencies in OU naming while we are updating our documentation to the new names.

To get started with AWS Control Tower from the AWS Management Console, go to the AWS Control Tower console, and select Set up landing zone in the top right. For additional information, you can read about planning your AWS Control Tower landing zone.

AWS Control Tower landing zone version 2.7

April 8, 2021

(Update required for AWS Control Tower landing zone to version 2.7. For information, see Update Your Landing Zone (p. 47))

With AWS Control Tower version 2.7, AWS Control Tower introduces four new mandatory preventative Log Archive controls that implement policy solely on AWS Control Tower resources. We have adjusted the guidance on four existing Log Archive controls from mandatory to elective, because they set policy for resources outside of AWS Control Tower. This control change and expansion provides the ability to separate Log Archive governance for resources within AWS Control Tower from governance of resources outside of AWS Control Tower.

The four changed controls can be used in conjunction with the new mandatory controls to provide governance to a broader set of AWS Log Archives. Existing AWS Control Tower environments will keep these four changed controls enabled automatically, for environment consistency; however, these elective controls now can be disabled. New AWS Control Tower environments must enable all elective controls. Existing environments must disable the formerly mandatory controls before adding encryption to Amazon S3 buckets that are not deployed by AWS Control Tower.

New mandatory controls:

- Disallow Changes to Encryption Configuration for AWS Control Tower Created S3 Buckets in Log Archive
- Disallow Changes to Logging Configuration for AWS Control Tower Created S3 Buckets in Log Archive
- Disallow Changes to Bucket Policy for AWS Control Tower Created S3 Buckets in Log Archive
- Disallow Changes to Lifecycle Configuration for AWS Control Tower Created S3 Buckets in Log Archive
Guidance changed from Mandatory to Elective:

- Disallow Changes to Encryption Configuration for all Amazon S3 Buckets [Previously: Enable Encryption at Rest for Log Archive]
- Disallow Changes to Logging Configuration for all Amazon S3 Buckets [Previously: Enable Access Logging for Log Archive]
- Disallow Changes to Bucket Policy for all Amazon S3 Buckets [Previously: Disallow Policy Changes to Log Archive]
- Disallow Changes to Lifecycle Configuration for all Amazon S3 Buckets [Previously: Set a Retention Policy for Log Archive]

AWS Control Tower version 2.7 includes changes to the AWS Control Tower landing zone blueprint that can cause incompatibility with previous versions after you upgrade to 2.7.

- In particular, AWS Control Tower version 2.7 enables `BlockPublicAccess` automatically on S3 buckets deployed by AWS Control Tower. You can turn this default off if your workload requires access across accounts. For more information about what happens with `BlockPublicAccess` enabled, see `Blocking public access to your Amazon S3 storage`.
- AWS Control Tower version 2.7 includes a requirement for HTTPS. All requests sent to S3 buckets deployed by AWS Control Tower must use secure socket layer (SSL). Only HTTPS requests are allowed to pass. If you use HTTP (without SSL) as an endpoint to send the requests, this change gives you an access denied error, which can potentially break your workflow. **This change cannot be reverted after the 2.7 update to your landing zone.**

  We recommend that you change your requests to use TLS instead of HTTP.

Three new AWS Regions available

April 8, 2021

(Update required for AWS Control Tower landing zone)

AWS Control Tower is available in three additional AWS Regions: Asia Pacific (Tokyo) Region, Asia Pacific (Seoul) Region, and Asia Pacific (Mumbai) Region. A landing zone update to version 2.7 is required for expanding governance into these Regions.

Your landing zone is not expanded automatically into these Regions when you perform the update to version 2.7, you must view and select them in the Regions table for inclusion.

Govern selected Regions only

February 19, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower Region selection provides better ability to manage the geographical footprint of your AWS Control Tower resources. To expand the number of Regions in which you host AWS resources or workloads – for compliance, regulatory, cost, or other reasons – you can now select the additional Regions to govern.

Region selection is available when you set up a new landing zone or update your AWS Control Tower landing zone version. When you use Account Factory to create a new account or enroll a pre-existing member account, or when you use `Extend Governance` to enroll accounts in a pre-existing organizational unit, AWS Control Tower deploys its governance capabilities of centralized logging, monitoring, and controls in your chosen Regions in the accounts. For more information about selecting Regions, see `Configure your AWS Control Tower Regions (p. 98)`.
AWS Control Tower now extends governance to existing OUs in your AWS organizations

January 28, 2021

(No update required for AWS Control Tower landing zone)

Extend governance to existing organizational units (OUs) (those not in AWS Control Tower) from within the AWS Control Tower console. With this feature, you can bring top-level OUs and included accounts under AWS Control Tower governance. For information about extending governance to an entire OU, see Register an existing organizational unit with AWS Control Tower (p. 190).

When you register an OU, AWS Control Tower performs a series of checks to ensure successful extension of governance and enrollment of accounts within the OU. For more information about common issues associated with the initial registration of an OU, see Common causes of failure during registration or re-registration (p. 192).

You can also visit the AWS Control Tower product webpage or visit YouTube to watch this video about getting started with AWS Control Tower for AWS Organizations.

AWS Control Tower provides bulk account updates

January 28, 2021

(No update required for AWS Control Tower landing zone)

With the bulk update feature, you can now update all accounts in a registered AWS Organizations organizational unit (OU) containing up to 300 accounts, with a single click, from the AWS Control Tower dashboard. This is particularly useful in cases where you update your AWS Control Tower landing zone and must also update your enrolled accounts to align them to the current landing zone version.

This feature also helps you keep your accounts up to date when you update your AWS Control Tower landing zone to expand to new regions, or when you want to re-register an OU to ensure that all accounts in that OU have the latest controls applied. Bulk account update eliminates the need to update one account at a time or use an external script to perform the update on multiple accounts.

For information about updating a landing zone, see Update Your Landing Zone (p. 47).

For information about registering or re-registering an OU, see Register an existing organizational unit with AWS Control Tower (p. 190).

January - December 2020

In 2020, AWS Control Tower released the following updates:

- AWS Control Tower console now links to external AWS Config rules (p. 1486)
- AWS Control Tower now available in additional Regions (p. 1486)
- Guardrail update (p. 1486)
- AWS Control Tower console shows more detail about OUs and accounts (p. 1487)
- Use AWS Control Tower to set up new multi-account AWS environments in AWS Organizations (p. 1487)
- Customizations for AWS Control Tower solution (p. 1488)
- General availability of AWS Control Tower version 2.3 (p. 1488)
AWS Control Tower console now links to external AWS Config rules

December 29, 2020

(Update required for AWS Control Tower landing zone to version 2.6. For information, see Update Your Landing Zone (p. 47))

AWS Control Tower now includes an organization-level aggregator, which assists in detecting external AWS Config rules. This provides you with visibility in the AWS Control Tower console to see the existence of externally created AWS Config rules in addition to those AWS Config rules created by AWS Control Tower. The aggregator allows AWS Control Tower to detect external rules and provide a link to the AWS Config console without the need for AWS Control Tower to gain access to unmanaged accounts.

With this feature, you now have a consolidated view of detective controls applied to your accounts so you can track compliance and determine if you need additional controls for your account. For information, see How AWS Control Tower aggregates AWS Config rules in unmanaged OUs and accounts (p. 91).

AWS Control Tower now available in additional Regions

November 18, 2020

(Update required for AWS Control Tower landing zone to version 2.5. For information, see Update Your Landing Zone (p. 47))

AWS Control Tower is now available in 5 additional AWS Regions:

- Asia Pacific (Singapore) Region
- Europe (Frankfurt) Region
- Europe (London) Region
- Europe (Stockholm) Region
- Canada (Central) Region

The addition of these 5 AWS Regions is the only change introduced for version 2.5 of AWS Control Tower.

AWS Control Tower is also available in US East (N. Virginia) Region, US East (Ohio) Region, US West (Oregon) Region, Europe (Ireland) Region, and Asia Pacific (Sydney) Region. With this launch AWS Control Tower is now available in 10 AWS Regions.

This landing zone update includes all Regions listed and cannot be undone. After updating your landing zone to version 2.5, you must manually update all enrolled accounts for AWS Control Tower to govern in the 10 supported AWS Regions. For information, see Configure your AWS Control Tower Regions (p. 98).

Guardrail update

October 8, 2020
AWS Control Tower User Guide
AWS Control Tower console shows more detail about OUs and accounts

(No update required for AWS Control Tower landing zone)

An updated version has been released for the mandatory control AWS-GR_IAM_ROLE_CHANGE_PROHIBITED.

This change to the control is required because accounts that are being enrolled automatically into AWS Control Tower must have the AWSControlTowerExecution role enabled. The previous version of the control prevents this role from being created.

For more information, see Control update (p. 226) in the AWS Control Tower User Guide Control reference.

**AWS Control Tower console shows more detail about OUs and accounts**

July 22, 2020

(No update required for AWS Control Tower landing zone)

You can view your organizations and accounts that are not enrolled in AWS Control Tower, alongside organizations and accounts that are enrolled.

Within the AWS Control Tower console, you can view more detail about your AWS accounts and organizational units (OUs). The Accounts page now lists all accounts in your organization, regardless of OU or enrollment status in AWS Control Tower. You can now search, sort, and filter across all tables.

**Use AWS Control Tower to set up new multi-account AWS environments in AWS Organizations**

April 22, 2020

(No update required for AWS Control Tower landing zone)

AWS Organizations customers can now use AWS Control Tower to manage newly created organizational units (OUs) and accounts by taking advantage of these new capabilities:

- Existing AWS Organizations customers can now set up a new landing zone for new organizational units (OUs) in their existing management account. You can create new OUs in AWS Control Tower and create new accounts in those OUs with AWS Control Tower governance.
- AWS Organizations customers can enroll existing accounts using the account enrollment process or through scripting.

AWS Control Tower provides an orchestration service that uses other AWS services. It's designed for organizations with multiple accounts and teams who are looking for the easiest way to set up their new or existing multi-account AWS environment and govern at scale. With an organization governed by AWS Control Tower, cloud administrators know that accounts in the organization are compliant with established policies. Builders benefit because they can provision new AWS accounts quickly, without undue concerns about compliance.

For information about setting up a landing zone, see Plan your AWS Control Tower landing zone (p. 33). You can also visit the AWS Control Tower product webpage or visit YouTube to watch this video about getting started with AWS Control Tower for AWS Organizations.

In addition to this change, the Quick account provisioning capability in AWS Control Tower was renamed to Enroll account. It now permits enrollment of existing AWS accounts as well as creation of new accounts. For more information, see Enroll an existing account (p. 110).
Customizations for AWS Control Tower solution

March 17, 2020

(No update required for AWS Control Tower landing zone)

AWS Control Tower now includes a new reference implementation that makes it easy for you to apply custom templates and policies to your AWS Control Tower landing zone.

With customizations for AWS Control Tower, you can use AWS CloudFormation templates to deploy new resources to existing and new accounts within your organization. You can also apply custom service control policies (SCPs) to those accounts in addition to the SCPs already provided by AWS Control Tower. Customizations for AWS Control Tower pipeline integrate with AWS Control Tower lifecycle events and notifications (Lifecycle Events in AWS Control Tower (p. 1416)) to ensure that resource deployments stay in sync with your landing zone.

The deployment documentation for this AWS Control Tower solution architecture is available through the AWS Solutions web page.

General availability of AWS Control Tower version 2.3

March 5, 2020

(Update required for AWS Control Tower landing zone. For information, see Update Your Landing Zone (p. 47).)

AWS Control Tower is now available in the Asia Pacific (Sydney) AWS Region, in addition to the US East (Ohio), US East (N. Virginia), US West (Oregon), and Europe (Ireland) Regions. The addition of the Asia Pacific (Sydney) Region is the only change introduced for version 2.3 of AWS Control Tower.

If you have not used AWS Control Tower previously, you can launch it today in any of the supported Regions. If you are already using AWS Control Tower and want to extend its governance features to the Asia Pacific (Sydney) Region in your accounts, go to the Settings page in your AWS Control Tower dashboard. From there, update your landing zone to the latest release. Then, update your accounts individually.

**Note**

Updating your landing zone does not automatically update your accounts. If you have more than a few accounts, the required updates can be time-consuming. For that reason, we recommend that you avoid expanding your AWS Control Tower landing zone into Regions in which you do not require your workloads to run.

For information about the expected behavior of detective controls as a result of a deployment to a new Region, see Configure your AWS Control Tower Regions (p. 98).

Single-step account provisioning in AWS Control Tower

March 2, 2020

(No update required for AWS Control Tower landing zone)

AWS Control Tower now supports single-step account provisioning through the AWS Control Tower console. This feature allows you to provision new accounts from within the AWS Control Tower console.

To use the simplified form, navigate to Account Factory in the AWS Control Tower console and then choose Quick account provisioning. AWS Control Tower assigns the same email address to the
AWS Control Tower decommissioning tool

February 28, 2020
(No update required for AWS Control Tower landing zone)

AWS Control Tower now supports an automated decommissioning tool to assist you in cleaning up resources allocated by AWS Control Tower. If you no longer intend to use AWS Control Tower for your enterprise, or if you require a major redeployment of your organizational resources, you may want to clean up the resources created when you initially set up your landing zone.

To decommission your landing zone by using a process that is mostly automated, contact AWS Support to get assistance with the additional steps that are required. For more information about decommissioning, see Walkthrough: Decommission an AWS Control Tower Landing Zone (p. 1434).

AWS Control Tower lifecycle event notifications

January 22, 2020
(No update required for AWS Control Tower landing zone)

AWS Control Tower announces the availability of lifecycle event notifications. A lifecycle event (p. 1416) marks the completion of an AWS Control Tower action that can change the state of resources such as organizational units (OUs), accounts, and controls that are created and managed by AWS Control Tower. Lifecycle events are recorded as AWS CloudTrail events and delivered to Amazon EventBridge as events.

AWS Control Tower records lifecycle events at the completion of the following actions that can be performed using the service: creating or updating a landing zone; creating or deleting an OU; enabling or disabling a control on an OU; and using account factory to create a new account or to move an account to another OU.

AWS Control Tower uses multiple AWS services to build and govern a best practices multi-account AWS environment. It can take several minutes for an AWS Control Tower action to complete. You can track lifecycle events in the CloudTrail logs to verify if the originating AWS Control Tower action completed successfully. You can create an EventBridge rule to notify you when CloudTrail records a lifecycle event or to automatically trigger the next step in your automation workflow.

January - December 2019

From January 1 through December 31, 2019, AWS Control Tower released the following updates:

- General availability of AWS Control Tower version 2.2 (p. 1490)
- New elective controls in AWS Control Tower (p. 1490)
General availability of AWS Control Tower version 2.2

November 13, 2019

Update required for AWS Control Tower landing zone. For information, see Update Your Landing Zone (p. 47).

AWS Control Tower version 2.2 provides three new preventive controls that prevent drift in accounts:

- **Disallow Changes to Amazon CloudWatch Logs Log Groups set up by AWS Control Tower** (p. 218)
- **Disallow Deletion of AWS Config Aggregation Authorizations Created by AWS Control Tower** (p. 219)
- **Disallow Deletion of Log Archive** (p. 219)

A control is a high-level rule that provides ongoing governance for your overall AWS environment. When you create your AWS Control Tower landing zone, the landing zone and all the organizational units (OUs), accounts, and resources are compliant with the governance rules enforced by your chosen controls. As you and your organization members use the landing zone, changes (accidental or intentional) in this compliance status may occur. Drift detection helps you identify resources that need changes or configuration updates to resolve the drift. For more information, see Detect and resolve drift in AWS Control Tower (p. 169).

New elective controls in AWS Control Tower

September 05, 2019

(No update required for AWS Control Tower landing zone)

AWS Control Tower now includes the following four new elective controls:

- **Disallow Delete Actions on Amazon S3 Buckets Without MFA** (p. 1372)
- **Disallow Changes to Replication Configuration for Amazon S3 Buckets** (p. 1372)
- **Disallow Actions as a Root User** (p. 1361)
- **Disallow Creation of Access Keys for the Root User** (p. 1361)

A control is a high-level rule that provides ongoing governance for your overall AWS environment. Guardrails enable you to express your policy intentions. For more information, see About controls in AWS Control Tower (p. 196).

New detective controls in AWS Control Tower

August 25, 2019

(No update required for AWS Control Tower landing zone)

AWS Control Tower now includes the following eight new detective controls:

- **Detect Whether Versioning for Amazon S3 Buckets is Enabled** (p. 1374)
- **Detect Whether MFA is Enabled for AWS IAM Users of the AWS Console** (p. 1374)
A control is a high-level rule that provides ongoing governance for your overall AWS environment. A detective control detects noncompliance of resources within your accounts, such as policy violations, and provides alerts through the dashboard. For more information, see About controls in AWS Control Tower (p. 196).

AWS Control Tower accepts email addresses for shared accounts with different domains than the management account

August 01, 2019

(No update required for AWS Control Tower landing zone)

In AWS Control Tower, you can now submit email addresses for shared accounts (log archive and audit member) and child accounts (vended using account factory) whose domains are different from the management account's email address. This feature is available only when you create a new landing zone and when you provision new child accounts.

General availability of AWS Control Tower version 2.1

June 24, 2019

(Update required for AWS Control Tower landing zone. For information, see Update Your Landing Zone (p. 47).)

AWS Control Tower is now generally available and supported for production use. AWS Control Tower is intended for organizations with multiple accounts and teams who are looking for the easiest way to set up their new multi-account AWS environment and govern at scale. With AWS Control Tower, you can help make sure that accounts in your organization are compliant with established policies. End users on distributed teams can provision new AWS accounts quickly.

Using AWS Control Tower, you can set up a landing zone (p. 16) that employs best practices such as configuring a multi-account structure using AWS Organizations, managing user identities and federated access with AWS IAM Identity Center, enabling account provisioning through Service Catalog, and creating a centralized log archive using AWS CloudTrail and AWS Config.

For ongoing governance, you can enable pre-configured controls, which are clearly defined rules for security, operations, and compliance. Guardrails help prevent deployment of resources that don’t conform to policies and continuously monitor deployed resources for nonconformance. The AWS Control Tower dashboard provides centralized visibility into an AWS environment including accounts provisioned, controls enabled, and the compliance status of accounts.

You can set up a new multi-account environment with a single click in the AWS Control Tower console. There are no additional charges or upfront commitments to use AWS Control Tower. You pay only for those AWS services that you enabled to set up a landing zone and implement selected controls.
The following table describes important changes to the *AWS Control Tower User Guide*. For notifications about documentation updates, you can subscribe to the RSS feed.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Control Tower available in Asia Pacific (Melbourne) AWS Region</td>
<td>Available in Asia Pacific (Melbourne) Region.</td>
<td>November 3, 2023</td>
</tr>
<tr>
<td>New control API available</td>
<td>AWS Control Tower released a new control API.</td>
<td>October 14, 2023</td>
</tr>
<tr>
<td>AWS Control Tower launches new controls</td>
<td>AWS Control Tower released new proactive and detective controls.</td>
<td>October 5, 2023</td>
</tr>
<tr>
<td>AWS Control Tower reports drift from disabling trusted access</td>
<td>AWS Control Tower notifies customers when drift occurs, if customers turn off trusted access to AWS Control Tower in AWS Organizations.</td>
<td>September 21, 2023</td>
</tr>
<tr>
<td>AWS Control Tower available in four additional AWS Regions</td>
<td>Available in Asia Pacific (Hyderabad), Europe (Spain and Zurich), and Middle East (UAE).</td>
<td>September 13, 2023</td>
</tr>
<tr>
<td>AWS Control Tower available in Tel Aviv Region</td>
<td>AWS Control Tower is available in the Tel Aviv Region, il-central-1.</td>
<td>August 28, 2023</td>
</tr>
<tr>
<td>AWS Control Tower launches 28 new proactive controls</td>
<td>AWS Control Tower released 28 new proactive controls.</td>
<td>July 24, 2023</td>
</tr>
<tr>
<td>AWS Control Tower deprecates 2 controls</td>
<td>AWS Control Tower will remove two controls from the controls library, effective August 18, 2023.</td>
<td>July 18, 2023</td>
</tr>
<tr>
<td>AWS Control Tower landing zone 3.2 available</td>
<td>AWS Control Tower landing zone version 3.2 is available.</td>
<td>June 16, 2023</td>
</tr>
<tr>
<td>AWS Control Tower handles accounts based on ID</td>
<td>AWS Control Tower tracks the AWS account ID, rather than the account's email address.</td>
<td>June 14, 2023</td>
</tr>
<tr>
<td>Additional Security Hub detective controls available</td>
<td>AWS Control Tower adds ten new controls to the controls library, for the Security Hub Service-Managed Standard: AWS Control Tower.</td>
<td>June 12, 2023</td>
</tr>
<tr>
<td>AWS Control Tower publishes control metadata tables (p. 1492)</td>
<td>AWS Control Tower now provides tables of control metadata as part of the published documentation.</td>
<td>June 7, 2023</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Terraform support for Account Factory Customization (p. 1492)</td>
<td>Single-region support for Terraform open source blueprints in AFC.</td>
<td>June 6, 2023</td>
</tr>
<tr>
<td>AWS IAM self-management available for landing zone (p. 1492)</td>
<td>AWS Control Tower now supports customers in choosing their identity provider for a landing zone.</td>
<td>June 6, 2023</td>
</tr>
<tr>
<td>New role added (p. 1492)</td>
<td>AWS Control Tower added a new service-linked role, <code>AWSServiceRoleForAWSControlTower</code>, and associated policy, <code>AWSControlTowerAccountServiceRolePolicy</code>.</td>
<td>June 1, 2023</td>
</tr>
<tr>
<td>Mixed governance update (p. 1492)</td>
<td>Update to advise customers regarding mixed governance.</td>
<td>June 1, 2023</td>
</tr>
<tr>
<td>Additional proactive controls available (p. 1492)</td>
<td>New proactive controls assist you in governing your multi-account environment and meeting specific control objectives.</td>
<td>May 19, 2023</td>
</tr>
<tr>
<td>Seven additional Regions available (p. 1492)</td>
<td>AWS Control Tower is now available in seven additional AWS Regions: Northern California (San Francisco), Asia Pacific (Hong Kong, Jakarta, and Osaka), Europe (Milan), Middle East (Bahrain), and Africa (Cape Town).</td>
<td>April 19, 2023</td>
</tr>
<tr>
<td>Change to a managed policy (p. 1492)</td>
<td>We changed the <code>AWSControlTowerServiceRolePolicy</code> so that AWS Control Tower can call the EnableRegion, ListRegions, GetRegionOptStatus APIs that are implemented by the AWS Account Management service.</td>
<td>April 6, 2023</td>
</tr>
<tr>
<td>Account customization request tracing generally available (p. 1492)</td>
<td>AWS Control Tower now supports the ability to trace account customization requests using the Account Factory for Terraform (AFT) workflow.</td>
<td>February 16, 2023</td>
</tr>
<tr>
<td>IAM best practices update (p. 1492)</td>
<td>Updated guide to align with the IAM best practices recommendations. For more information, see Security best practices in IAM.</td>
<td>February 15, 2023</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>AWS Control Tower landing zone 3.1 available (p. 1492)</td>
<td>AWS Control Tower landing zone 3.1 is available.</td>
<td>February 9, 2023</td>
</tr>
<tr>
<td>Proactive controls generally available (p. 1492)</td>
<td>Proactive controls are launched from preview status to general availability.</td>
<td>January 24, 2023</td>
</tr>
<tr>
<td>Concurrent account operations (p. 1492)</td>
<td>AWS Control Tower now supports up to five (5) concurrent actions in account factory. You can create, update, or enroll up to five accounts at a time.</td>
<td>December 16, 2022</td>
</tr>
<tr>
<td>Proactive controls assist in resource provisioning (p. 1492)</td>
<td>AWS Control Tower now supports proactive controls, implemented through AWS CloudFormation hooks.</td>
<td>November 28, 2022</td>
</tr>
<tr>
<td>Account factory customization available (p. 1492)</td>
<td>AWS Control Tower now supports account provisioning with customizable account templates, called blueprints, directly from the AWS Control Tower console.</td>
<td>November 28, 2022</td>
</tr>
<tr>
<td>Compliance status viewable for all AWS Config rules (p. 1492)</td>
<td>AWS Control Tower now displays the compliance status of all AWS Config rules deployed into organizational units registered with AWS Control Tower.</td>
<td>November 18, 2022</td>
</tr>
<tr>
<td>Change to a managed policy (p. 1492)</td>
<td>We changed the <code>AWSControlTowerServiceRolePolicy</code> so that AWS Control Tower can assume the <code>AWSControlTowerBlueprintAccess</code> role, which is needed for Account Factory customizations.</td>
<td>October 28, 2022</td>
</tr>
<tr>
<td>APIs for controls, AWS CloudFormation resource (p. 1492)</td>
<td>AWS Control Tower now supports activation and deactivation of controls through a set of API calls, and a new AWS CloudFormation resource.</td>
<td>September 1, 2022</td>
</tr>
<tr>
<td>CfCT supports stack set deletion (p. 1492)</td>
<td>CfCT supports stack set deletion, by setting a parameter in the manifest file.</td>
<td>August 26, 2022</td>
</tr>
<tr>
<td>Customized log retention (p. 1492)</td>
<td>You can customize the retention policy for Amazon S3 buckets that store your AWS Control Tower CloudTrail logs, in increments of days or years, up to a maximum of 15 years.</td>
<td>August 15, 2022</td>
</tr>
<tr>
<td><strong>Role drift repair available (p. 1492)</strong></td>
<td>AWS Control Tower supports repair for role drift, without a full repair of the landing zone.</td>
<td>August 11, 2022</td>
</tr>
<tr>
<td><strong>Version 3.0 available (p. 1492)</strong></td>
<td>AWS Control Tower landing zone version 3.0 changes from account-based AWS CloudTrail trails to organization-based trails, and it updates the managed policy to enable organization-level trails. It enables you to aggregate AWS Config information in your home Region only. Version 3.0 also includes an update to the Region deny control, and two new detective controls.</td>
<td>July 29, 2022</td>
</tr>
<tr>
<td><strong>The Organization page combines views of OUs and accounts (p. 1492)</strong></td>
<td>The new Organization page in AWS Control Tower shows a hierarchical view of all Organizational units (OUs) and accounts.</td>
<td>July 18, 2022</td>
</tr>
<tr>
<td><strong>Change to a managed policy (p. 1492)</strong></td>
<td>We changed the AWSControlTowerServiceRolePolicy so that customers can have organization-level AWS CloudTrail trails to aggregate AWS CloudTrail logs.</td>
<td>June 20, 2022</td>
</tr>
<tr>
<td><strong>Easier enroll and update for member accounts (p. 1492)</strong></td>
<td>AWS Control Tower now gives you the capability to enroll and update member accounts individually, from within your landing zone. Each account shows when it is available for an update. We separated the Enroll account button from the Create account workflow in Account Factory.</td>
<td>May 31, 2022</td>
</tr>
<tr>
<td><strong>AFT supports customization for shared accounts (p. 1492)</strong></td>
<td>AWS Control Tower Account Factory for Terraform now supports customization for the AWS Control Tower management account, log archive, and audit accounts.</td>
<td>May 27, 2022</td>
</tr>
<tr>
<td><strong>Concurrent operations for all optional controls (p. 1492)</strong></td>
<td>AWS Control Tower now allows you to apply and remove optional preventive guardails concurrently, as well as detective controls.</td>
<td>May 18, 2022</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>Existing security and logging accounts (p. 1492)</td>
<td>AWS Control Tower now supports the ability to bring existing security and logging accounts, rather than creating new ones during landing zone setup.</td>
<td>May 16, 2022</td>
</tr>
<tr>
<td>Version 2.9 available (p. 1492)</td>
<td>AWS Control Tower landing zone version 2.9 updates the notification forwarder Lambda to use the Python version 3.9 runtime.</td>
<td>April 22, 2022</td>
</tr>
<tr>
<td>Updated support for AWS best practices, version 2.8 available (p. 1492)</td>
<td>AWS Control Tower landing zone version 2.8 provides additional support to ensure that your workloads and AWS accounts are in alignment with AWS best practices.</td>
<td>February 10, 2022</td>
</tr>
<tr>
<td>Region deny control (p. 1492)</td>
<td>AWS Control Tower now includes a control that helps you restrict access to AWS Regions, to address compliance and regulatory concerns.</td>
<td>November 30, 2021</td>
</tr>
<tr>
<td>Data residency controls (p. 1492)</td>
<td>AWS Control Tower now support controls that help you manage data residency with granular control.</td>
<td>November 30, 2021</td>
</tr>
<tr>
<td>AWS Control Tower Account factory for Terraform (p. 1492)</td>
<td>AWS Control Tower now supports Terraform for automated account provisioning and updating.</td>
<td>November 29, 2021</td>
</tr>
<tr>
<td>New lifecycle event available (p. 1492)</td>
<td>The PrecheckOrganizationalUnit event logs whether any resources block the Extend governance task from success, including resources in nested OUs.</td>
<td>November 18, 2021</td>
</tr>
<tr>
<td>Nested OUs available (p. 1492)</td>
<td>AWS Control Tower now enables your landing zone to contain nested OU structures.</td>
<td>November 16, 2021</td>
</tr>
<tr>
<td>Detective control concurrency (p. 1492)</td>
<td>AWS Control Tower detective controls now support concurrent enable and disable operations.</td>
<td>November 5, 2021</td>
</tr>
<tr>
<td>Two new regions available (p. 1492)</td>
<td>AWS Control Tower is now available in two new AWS Regions, Europe (Paris) Region and South America (São Paulo) Region.</td>
<td>July 29, 2021</td>
</tr>
<tr>
<td><strong>Region deselection (p. 1492)</strong></td>
<td>You can deselect AWS Regions that you no longer wish to govern through AWS Control Tower.</td>
<td>July 29, 2021</td>
</tr>
<tr>
<td><strong>KMS keys available (p. 1492)</strong></td>
<td>You can optionally create or choose KMS keys that you manage, to encrypt your data and resources.</td>
<td>July 28, 2021</td>
</tr>
<tr>
<td><strong>Change to a managed policy (p. 1492)</strong></td>
<td>We changed the AWSControlTowerServiceRolePolicy so that customers can use their own KMS encryption keys for AWS CloudTrail logs.</td>
<td>July 28, 2021</td>
</tr>
<tr>
<td><strong>Control names changed, functionality unchanged (p. 1492)</strong></td>
<td>Certain control names and descriptions were updated to better reflect the policy intentions of the control, with no change in functionality.</td>
<td>July 26, 2021</td>
</tr>
<tr>
<td><strong>Automated scans of managed SCPs (p. 1492)</strong></td>
<td>AWS Control Tower performs daily automated scans of managed SCPs to check for drift.</td>
<td>May 11, 2021</td>
</tr>
<tr>
<td><strong>Customized names for OUs and accounts (p. 1492)</strong></td>
<td>AWS Control Tower allows you to provide customized names during the landing zone setup process, for essential OUs and accounts, without creating drift.</td>
<td>April 16, 2021</td>
</tr>
<tr>
<td><strong>Decommissioning a landing zone is self-service (p. 1492)</strong></td>
<td>AWS Control Tower now allows you to decommission a landing zone without contacting AWS Support. Decommissioning is a semi-automated process that cannot be undone. It is not the same as deleting all AWS Control Tower resources manually.</td>
<td>April 9, 2021</td>
</tr>
<tr>
<td><strong>Three additional Regions (p. 1492)</strong></td>
<td>AWS Control Tower is now available in three additional AWS Regions: Asia Pacific (Tokyo) Region, Asia Pacific (Seoul) Region, and Asia Pacific (Mumbai) Region.</td>
<td>April 8, 2021</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td><strong>New Log Archive controls, landing zone version 2.7 available (p. 1492)</strong></td>
<td>Four new Log Archive controls provide Log Archive governance over AWS Control Tower resources, separately from governance of resources outside of AWS Control Tower. Guidance on four existing controls has changed from mandatory to elective. Version 2.7 of the AWS Control Tower landing zone includes a requirement for HTTPS, which cannot be undone after you update.</td>
<td>April 8, 2021</td>
</tr>
<tr>
<td><strong>Region selection (p. 1492)</strong></td>
<td>AWS Control Tower Region selection provides better ability to manage the geographical footprint of your AWS Control Tower resources. To expand the number of Regions in which you host AWS resources or workloads – for compliance, regulatory, cost, or other reasons – you can now select the additional Regions to govern.</td>
<td>February 19, 2021</td>
</tr>
<tr>
<td><strong>Register an OU and govern all of its accounts with AWS Control Tower at one time (p. 1492)</strong></td>
<td>AWS Control Tower adds the capability to register an OU, which is a way to bring multiple accounts into governance at the same time.</td>
<td>January 28, 2021</td>
</tr>
<tr>
<td><strong>Multiple account updates in registered OUs (p. 1492)</strong></td>
<td>You can now update all accounts in any registered AWS Organizations organizational unit (OU) containing up to 300 accounts, with a single click, from the AWS Control Tower dashboard. The multiple account update feature, also referred to as bulk update, eliminates the need to update one account at a time, or to use an external script to perform the update on multiple accounts together.</td>
<td>January 28, 2021</td>
</tr>
<tr>
<td><strong>New role for aggregating unmanaged OUs and accounts (p. 1492)</strong></td>
<td>A new role assists in detecting external AWS Config rules, so AWS Control Tower does not need to gain access to unmanaged accounts.</td>
<td>December 29, 2020</td>
</tr>
<tr>
<td>AWS Control Tower is available in more AWS Regions. (p. 1492)</td>
<td>AWS Control Tower is now available to be deployed in the Asia Pacific (Singapore) Region, Europe (Frankfurt) Region, Europe (London) Region, Europe (Stockholm) Region, and Canada (Central) Region. With this launch AWS Control Tower is now available in 10 AWS Regions. This landing zone update includes all Regions listed, and it cannot be undone. After updating your landing zone to version 2.5, you must manually update all enrolled accounts for AWS Control Tower to govern in the 10 supported AWS Regions.</td>
<td>November 18, 2020</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Control update (p. 1492)</td>
<td>An updated version has been released for the mandatory control AWS-GR_IAM_ROLE_CHANGE_PROHIBITED. The updated control allows easier automated enrollment of accounts.</td>
<td>October 8, 2020</td>
</tr>
<tr>
<td>Related information page is now available for AWS Control Tower (p. 1492)</td>
<td>The related information page makes it easier to find common tasks that may be helpful after setting up your AWS Control Tower landing zone.</td>
<td>September 18, 2020</td>
</tr>
<tr>
<td>AWS Control Tower console shows more detail about OUs and accounts. (p. 1492)</td>
<td>Within the AWS Control Tower console, you can view more detail about your AWS accounts and organizational units (OUs). The 'Accounts' page now lists all accounts in your organization, regardless of OU or enrollment status in AWS Control Tower. You can now search, sort, and filter across all tables.</td>
<td>July 22, 2020</td>
</tr>
<tr>
<td>AWS Control Tower allows existing organizations to set up a landing zone (p. 1492)</td>
<td>You can now launch a landing zone for AWS Control Tower in an existing organization, to bring the organization into governance. The <strong>Quick account provisioning</strong> capability in AWS Control Tower was renamed to <strong>Enroll account</strong> and it now permits enrollment of existing AWS accounts as well as creation of new accounts.</td>
<td>April 16, 2020</td>
</tr>
<tr>
<td>AWS Control Tower is now available in Asia Pacific (p. 1492)</td>
<td>AWS Control Tower is now available to be deployed in the Asia Pacific (Sydney) AWS Region. This release requires manual updates to vended accounts, update only if you plan to run workloads in Asia Pacific (Sydney).</td>
<td>March 3, 2020</td>
</tr>
<tr>
<td>Decommissioning an AWS Control Tower landing zone is possible (p. 1492)</td>
<td>AWS Support can help you permanently decommission a landing zone through a mostly automated process that preserves your organizations, although some manual cleanup is required.</td>
<td>February 27, 2020</td>
</tr>
<tr>
<td>Quick account provisioning is available in AWS Control Tower (p. 1492)</td>
<td>Quick account provisioning makes it easier to launch new member accounts when your landing zone is up to date, with the Enroll account feature.</td>
<td>February 20, 2020</td>
</tr>
<tr>
<td>Lifecycle events are tracked in AWS Control Tower (p. 1492)</td>
<td>Lifecycle events provide additional details for certain AWS Control Tower events, to make some workflow automation easier.</td>
<td>December 12, 2019</td>
</tr>
<tr>
<td>Settings and Activities pages are available for AWS Control Tower (p. 1492)</td>
<td>The Settings and Activities pages make it easier to update your landing zone and to view logged events.</td>
<td>November 30, 2019</td>
</tr>
<tr>
<td>Additional preventive controls are available for AWS Control Tower (p. 1492)</td>
<td>Preventive controls in AWS Control Tower keep your organization and resources aligned with your environment.</td>
<td>September 6, 2019</td>
</tr>
<tr>
<td>Additional detective controls are available for AWS Control Tower (p. 1492)</td>
<td>Detective controls in AWS Control Tower give information about the state of your organization and resources.</td>
<td>August 27, 2019</td>
</tr>
<tr>
<td>AWS Control Tower is now generally available (p. 1492)</td>
<td>AWS Control Tower is a service that offers the easiest way to set up and govern your multi-account AWS environment at scale.</td>
<td>June 24, 2019</td>
</tr>
</tbody>
</table>
Tables of control metadata

This section contains tables that show the metadata for controls. Remember that each control has a unique API identifier for each Region in which AWS Control Tower is available. When you are working with the control APIs, provide the identifier for the Region in which you are making the API call.

The identifiers for mandatory controls are not shown in this table, because those controls cannot be turned on or turned off.

### AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
<td>• CIS AWS Benchmark</td>
<td>• Encrypt data at rest</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-9</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.5.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Asia Pacific (Hong Kong)| arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>east-1:control/</td>
<td></td>
<td></td>
<td>AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_AUDIT_BUCKET_ENCRYPTION_ENABLED</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED | • CIS AWS Benchmark | Establish logging and monitoring | • US East (N. Virginia)
|                     | 1.4 3.6   |                   | arn:aws:controltower:us-east-1::control/|
|                     | 1.4 3.6   |                   | AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED|
|                     | 1.4 3.6   |                   | • US East (Ohio)
|                     | 1.4 3.6   |                   | arn:aws:controltower:us-east-2::control/|
|                     | 1.4 3.6   |                   | AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED|
|                     | 1.4 3.6   |                   | • US West (Oregon)
|                     | 1.4 3.6   |                   | arn:aws:controltower:us-west-2::control/|
|                     | 1.4 3.6   |                   | AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED|
|                     | 1.4 3.6   |                   | • Canada (Central)
|                     | 1.4 3.6   |                   | arn:aws:controltower:ca-central-1::control/|
|                     | 1.4 3.6   |                   | AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED|
|                     | 1.4 3.6   |                   | • Asia Pacific (Sydney)
|                     | 1.4 3.6   |                   | arn:aws:controltower:ap-southeast-2::control/|
|                     | 1.4 3.6   |                   | AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED|
|                     | 1.4 3.6   |                   | • Asia Pacific (Singapore)
<p>|                     | 1.4 3.6   |                   | arn:aws:controltower:ap-southeast-1::control/|
|                     | 1.4 3.6   |                   | AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED|
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 AC-2(4) |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 AC-4(26) |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 AC-6(9) |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 AU-10 |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 AU-12 |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 AU-2 |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 AU-3 |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 AU-6(3) |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 AU-6(4) |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 CA-7 |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 SC-7(9) |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 SI-3(8) |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 SI-4(20) |
|                     | 1.4 3.6   |                   | • NIST 800-53 Rev 5 SI-7(8) |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/AWS-GR_AUDIT_BUCKET_LOGGING_ENABLED</td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3(8)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SA-8(19)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
<td></td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/</td>
<td>AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
<td>AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/</td>
<td>AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED | | | • US West (N. California)  
arn:aws:controltower:us-west-1::control/  
AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED |
| | | | • Asia Pacific (Hong Kong)  
arn:aws:controltower:ap-east-1::control/  
AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED |
| | | | • Asia Pacific (Jakarta)  
arn:aws:controltower:ap-southeast-3::control/  
AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED |
| | | | • Asia Pacific (Osaka)  
arn:aws:controltower:ap-northeast-3::control/  
AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED |
| | | | • Europe (Milan)  
arn:aws:controltower:eu-south-1::control/  
AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED |
| | | | • Africa (Cape Town)  
arn:aws:controltower:af-south-1::control/  
AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED |
| | | | • Middle East (Bahrain)  
arn:aws:controltower:me-south-1::control/  
AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED |
| | | | • Israel (Tel Aviv)  
arn:aws:controltower:il-central-1::control/  
AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED |
| | | | • Europe (Zurich)  
arn:aws:controltower:eu-central-2::control/  
AWS-GR_AUDIT_BUCKET_POLICY_CHANGES_PROHIBITED |
### AWS-GR_AUDIT_BUCKET_RETENTION_POLICY

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
• US East (Ohio) arn:aws:controltower:us-east-2::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY  
• US West (Oregon) arn:aws:controltower:us-west-2::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY  
• Canada (Central) arn:aws:controltower:ca-central-1::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:sa-east-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• US West (N. California)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-east-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
<td>AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
<td>AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:af-south-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:me-south-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:il-central-1::control/</td>
<td>AWS-GR_AUDIT_BUCKET_RETENTION_POLICY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>• Europe (Spain)</td>
<td>• Europe (Spain)</td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td>• Asia Pacific (Melbourne)</td>
<td>• Asia Pacific (Melbourne)</td>
<td>• Asia Pacific (Melbourne)</td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td>• US East (N. Virginia)</td>
<td>• US East (N. Virginia)</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td>• US East (Ohio)</td>
<td>• US West (Oregon)</td>
<td>• US West (Oregon)</td>
<td>• US West (Oregon)</td>
</tr>
</tbody>
</table>

**AWS-GR_AUTOSCALING_LAUNCH_CONFIG_PUBLIC_IP_DISABLED**

**AWS-GR_AUTOSCALING_LAUNCH_CONFIG_PUBLIC_IP_DISABLED**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td>• Limit network access</td>
<td>• Limit network access</td>
</tr>
</tbody>
</table>

**AWS-GR_AUDIT_BUCKET_RETENTION_POLICY**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (Zurich)</td>
<td>• Europe (Spain)</td>
<td>• Europe (Spain)</td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td>• Asia Pacific (Melbourne)</td>
<td>• Asia Pacific (Melbourne)</td>
<td>• Asia Pacific (Melbourne)</td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td>• US East (N. Virginia)</td>
<td>• US East (N. Virginia)</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td>• US East (Ohio)</td>
<td>• US West (Oregon)</td>
<td>• US West (Oregon)</td>
<td>• US West (Oregon)</td>
</tr>
</tbody>
</table>

**AWS-GR_AUDIT_BUCKET_RETENTION_POLICY**
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td>AWS-GR_AUTOSCALING_LAUNCH_CONFIG_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>nordest-1::control/aws-gr-autoscaling-launch-config_public_ip_disabled</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- NIST 800-53 Rev 5 AU-10
- NIST 800-53 Rev 5 AU-12
- NIST 800-53 Rev 5 AU-14(1)
- NIST 800-53 Rev 5 AU-2
- NIST 800-53 Rev 5 AU-3
- NIST 800-53 Rev 5 AU-6(3)
- NIST 800-53 Rev 5 AU-6(4)
- NIST 800-53 Rev 5 AU-9
- NIST 800-53 Rev 5 CA-7
- NIST 800-53 Rev 5 SA-8(22)

- Establish logging and monitoring

- US East (N. Virginia)
  - arn:aws:controltower:us-east-1::control/
    - AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS

- US East (Ohio)
  - arn:aws:controltower:us-east-2::control/
    - AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS

- US West (Oregon)
  - arn:aws:controltower:us-west-2::control/
    - AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS

- Canada (Central)
  - arn:aws:controltower:ca-central-1::control/
    - AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS

- Asia Pacific (Sydney)
  - arn:aws:controltower:ap-southeast-2::control/
    - AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS

- Asia Pacific (Singapore)
  - arn:aws:controltower:ap-southeast-1::control/
    - AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS

- Australian Commonwealth
  - au-3

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control identifier</td>
<td>AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS</td>
</tr>
<tr>
<td>Framework</td>
<td></td>
</tr>
<tr>
<td>Control objective</td>
<td></td>
</tr>
<tr>
<td>Control API identifiers, by Region</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.2</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.6</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.7</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.4.5</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.3</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.4</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.6</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.7</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.6.1</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.6.2</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.6.3</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.6.4</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.6.5</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.7.3</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.7.4</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.8.3</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.8.4</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.9.3</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.9.4</td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1517
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/</td>
<td>AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/</td>
<td>AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
<td>AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
<td>AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/</td>
<td>AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/</td>
<td>AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/</td>
<td>AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/</td>
<td>AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS</td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/</td>
<td>AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS</td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/</td>
<td>AWS-GR_DETECT_CLOUDTRAIL_ENABLED_ON_MEMBER_ACCOUNTS</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_DISALLOW_CROSS_REGION_NETWORKING | None | Limit network access | - US East (N. Virginia)  
  arn:aws:controltower:us-east-1::control/  
  AWS-GR_DISALLOW_CROSS_REGION_NETWORKING  
- US East (Ohio)  
  arn:aws:controltower:us-east-2::control/  
  AWS-GR_DISALLOW_CROSS_REGION_NETWORKING  
- US West (Oregon)  
  arn:aws:controltower:us-west-2::control/  
  AWS-GR_DISALLOW_CROSS_REGION_NETWORKING  
- Canada (Central)  
  arn:aws:controltower:ca-central-1::control/  
  AWS-GR_DISALLOW_CROSS_REGION_NETWORKING  
- Asia Pacific (Sydney)  
  arn:aws:controltower:ap-southeast-4::control/  
  AWS-GR_DISALLOW_CROSS_REGION_NETWORKING |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>southeast-2::control/</td>
<td>AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/ AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>aws:controltower:sa-east-1::control/</td>
<td>AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>aws:controltower:us-west-1::control/</td>
<td>AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>aws:controltower:ap-east-1::control/</td>
<td>AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>aws:controltower:ap-southeast-3::control/</td>
<td>AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>aws:controltower:ap-northeast-3::control/</td>
<td>AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>aws:controltower:eu-south-1::control/</td>
<td>AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>aws:controltower:af-south-1::control/</td>
<td>AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>aws:controltower:me-south-1::control/</td>
<td>AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>aws:controltower:il-central-1::control/</td>
<td>AWS-GR_DISALLOW_CROSS_REGION_NETWORKING</td>
</tr>
</tbody>
</table>
| • Europe (Zurich) | aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6</td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Limit network access</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2:control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1:control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1:control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1:control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2:control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1:control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1:control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2:control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1:control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
</tbody>
</table>

- NIST 800-53 Rev 5 SC-7(3)
- NIST 800-53 Rev 5 SC-7(4)
- NIST 800-53 Rev 5 SC-7(9)
- PCI DSS version 3.2.1 1.2.1
- PCI DSS version 3.2.1 1.3
- PCI DSS version 3.2.1 1.3.1
- PCI DSS version 3.2.1 1.3.2
- PCI DSS version 3.2.1 1.3.4
- PCI DSS version 3.2.1 1.3.6
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>west-3::control/</td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td>Amazon VPC</td>
<td></td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td>arn:aws:controltower:sa-east-1::control/</td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/</td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-east-1::control/</td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-1::control/</td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td>arn:aws:controltower:af-south-1::control/</td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td>arn:aws:controltower:me-south-1::control/</td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td>arn:aws:controltower:il-central-1::control/</td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPC_INTERNET_ACCESS</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_DISALLOW_VPC_INTERNET_ACCESS | None | Limit network access | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/  
AWS-GR_DISALLOW_VPC_INTERNET_ACCESS  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/  
AWS-GR_DISALLOW_VPC_INTERNET_ACCESS  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/  
AWS-GR_DISALLOW_VPC_INTERNET_ACCESS |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/ AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/ AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/ AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/ AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/ AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/ AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/ AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/ AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/ AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo) |           |                   | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeast-1::control/</td>
<td>AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/</td>
<td>AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/</td>
<td>AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/</td>
<td>AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/</td>
<td>AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
<td>AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
<td>AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/</td>
<td>AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/</td>
<td>AWS-GR_DISALLOW_VPN_CONNECTIONS</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
</tr>
<tr>
<td>AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td>• Improve availability</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>• US East (Ohio)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>ARN:AWS:CONTROLTOWER:AP-NORTHEAST-2::CONTROL/</td>
<td>AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>ARN:AWS:CONTROLTOWER:AP-NORTHEAST-1::CONTROL/</td>
<td>AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>ARN:AWS:CONTROLTOWER:EU-WEST-3::CONTROL/</td>
<td>AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>ARN:AWS:CONTROLTOWER:SA-EAST-1::CONTROL/</td>
<td>AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>ARN:AWS:CONTROLTOWER:AP-EAST-1::CONTROL/</td>
<td>AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>ARN:AWS:CONTROLTOWER:AP-SOUTHEAST-3::CONTROL/</td>
<td>AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>ARN:AWS:CONTROLTOWER:AP-NORTHEAST-3::CONTROL/</td>
<td>AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>ARN:AWS:CONTROLTOWER:EU-SOUTH-1::CONTROL/</td>
<td>AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
</tbody>
</table>
### AWS-GR_EBS_OPTIMIZED_INSTANCE

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Africa (Cape Town)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td><strong>Middle East (Bahrain)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td><strong>Israel (Tel Aviv)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td><strong>Europe (Zurich)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td><strong>Europe (Spain)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td><strong>Middle East (UAE)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td><strong>Asia Pacific (Melbourne)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/AWS-GR_EBS_OPTIMIZED_INSTANCE</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK | • NIST 800-53 Rev 5 AC-3  
• NIST 800-53 Rev 5 AC-3(7)  
• NIST 800-53 Rev 5 AC-4  
• NIST 800-53 Rev 5 AC-4(21)  
• NIST 800-53 Rev 5 AC-6  
• NIST 800-53 Rev 5 SC-7  
• NIST 800-53 Rev 5 SC-7(11)  
• NIST 800-53 Rev 5 SC-7(16)  
• NIST 800-53 Rev 5 SC-7(20)  
• NIST 800-53 Rev 5 SC-7(21)  
• NIST 800-53 Rev 5 SC-7(3)  
• NIST 800-53 Rev 5 SC-7(4)  
• NIST 800-53 Rev 5 SC-7(9)  
• PCI DSS version 3.2.1 1.2.1  
• PCI DSS version 3.2.1 1.3  
• PCI DSS version 3.2.1 1.3.1  
• PCI DSS version 3.2.1 1.3.2  
• PCI DSS version 3.2.1 1.3.4  
• PCI DSS version 3.2.1 1.3.6  
• PCI DSS version 3.2.1 2.2.2 | • Limit network access  
• Enforce least privilege | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/  
AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/  
AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/  
AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK |

1534
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/AWS-GR_EBS_SNAPSHOT_PUBLIC_RESTORABLE_CHECK</td>
</tr>
</tbody>
</table>
| Africa (Cape Town) |           |                  | arn:aws:controltower:af-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP | • NIST 800-53 Rev 5 | • Limit network access | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP 
• US East (Ohio) arn:aws:controltower:us-east-2::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
<td>east-2::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>Arn:aws:controltower:ap-northeast-2::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>Arn:aws:controltower:ap-northeast-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>Arn:aws:controltower:eu-west-3::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>Arn:aws:controltower:sa-east-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>Arn:aws:controltower:us-west-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>Arn:aws:controltower:ap-east-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>Arn:aws:controltower:eu-south-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>Arn:aws:controltower:af-south-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>Arn:aws:controltower:me-south-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>Arn:aws:controltower:il-1538</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
<td></td>
<td></td>
<td>central-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Zurich) arn:aws:controltower:eu-central-2::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Spain) arn:aws:controltower:eu-south-2::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Middle East (UAE) arn:aws:controltower:me-central-1::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/AWS-GR_EC2_INSTANCE_NO_PUBLIC_IP</td>
</tr>
</tbody>
</table>

AWS-GR_EC2_VOLUME_INUSE_CHECK

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
<td></td>
<td></td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
</tbody>
</table>
|                    |           |                  |     arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeast-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
<td>• Europe (Paris)</td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/AWS-GR_EC2_VOLUME_INUSE_CHECK</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS</td>
<td>• NIST 800-53 Rev 5</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>AC-3</td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td>AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS</td>
</tr>
<tr>
<td></td>
<td>AC-3(7)</td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>AC-4(21)</td>
<td></td>
<td>AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>AC-6</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7</td>
<td></td>
<td>AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(11)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(16)</td>
<td></td>
<td>AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(20)</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(21)</td>
<td></td>
<td>AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(3)</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(4)</td>
<td></td>
<td>AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(9)</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(9)</td>
<td></td>
<td>AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.2.1</td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3</td>
<td></td>
<td>AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.1</td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.2</td>
<td></td>
<td>AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.4</td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.6</td>
<td></td>
<td>AWS-GR_EKS_ENDPOINT_NO_PUBLIC_ACCESS</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.6</td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.6</td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
<td>• NIST 800-53 Rev 5</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td></td>
<td>AC-3</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td></td>
<td>AC-3(7)</td>
<td></td>
<td>• NIST 800-53 Rev 5</td>
</tr>
<tr>
<td></td>
<td>AC-3</td>
<td></td>
<td>• NIST 800-53 Rev 5</td>
</tr>
<tr>
<td></td>
<td>AC-4</td>
<td></td>
<td>• NIST 800-53 Rev 5</td>
</tr>
<tr>
<td></td>
<td>AC-4(21)</td>
<td></td>
<td>• NIST 800-53 Rev 5</td>
</tr>
<tr>
<td></td>
<td>AC-6</td>
<td></td>
<td>• NIST 800-53 Rev 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>west-2::control/AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eucentral-1::control/AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
</tbody>
</table>

1545
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/</td>
<td></td>
<td>AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/</td>
<td></td>
<td>AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/</td>
<td></td>
<td>AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/</td>
<td></td>
<td>AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/</td>
<td></td>
<td>AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/</td>
<td></td>
<td>AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/</td>
<td></td>
<td>AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/</td>
<td></td>
<td>AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/</td>
<td></td>
<td>AWS-GR_ELASTICSEARCH_IN_VPC_ONLY</td>
</tr>
</tbody>
</table>
### AWS-GR_EMR_MASTER_NO_PUBLIC_IP

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td>Limit network access</td>
<td><strong>US East (N. Virginia)</strong>&lt;br&gt;arn:aws:controltower:us-east-1::control/AWS-GR_EMR_MASTER_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td><strong>US East (Ohio)</strong>&lt;br&gt;arn:aws:controltower:us-east-2::control/AWS-GR_EMR_MASTER_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td><strong>US West (Oregon)</strong>&lt;br&gt;arn:aws:controltower:us-west-2::control/AWS-GR_EMR_MASTER_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td><strong>Canada (Central)</strong>&lt;br&gt;arn:aws:controltower:ca-central-1::control/AWS-GR_EMR_MASTER_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td><strong>Asia Pacific (Sydney)</strong>&lt;br&gt;arn:aws:controltower:ap-southeast-2::control/AWS-GR_EMR_MASTER_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td><strong>Asia Pacific (Singapore)</strong>&lt;br&gt;arn:aws:controltower:ap-southeast-1::control/AWS-GR_EMR_MASTER_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td><strong>Europe (Frankfurt)</strong>&lt;br&gt;arn:aws:controltower:eu-central-1::control/AWS-GR_EMR_MASTER_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td><strong>Europe (Ireland)</strong>&lt;br&gt;arn:aws:controltower:eu-west-1::control/AWS-GR_EMR_MASTER_NO_PUBLIC_IP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td><strong>Europe (London)</strong>&lt;br&gt;arn:aws:controltower:eu-west-2::control/AWS-GR_EMR_MASTER_NO_PUBLIC_IP</td>
</tr>
</tbody>
</table>
|                    | NIST 800-53 Rev 5 SC-7(21) | | **Europe (Stockholm)**<br>arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>north-1::control/</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td>AWS-GR_EMR_MASTER_NO_PUBLIC_IP</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:europe-west-3::control/</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:sao-east-1::control/</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-east-1::control/</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:me-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Arn:aws:controltower:me-south-1::control/</td>
</tr>
</tbody>
</table>
## AWS-GR_ENCRYPTED_VOLUMES

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| AWS-GR_ENCRYPTED_VOLUMES | • NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-3(6)  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-28  
• NIST 800-53 Rev 5 SC-28(1)  
• NIST 800-53 Rev 5 SC-7(10)  
• NIST 800-53 Rev 5 SI-7(6)  
• PCI DSS version 3.2.1 2.2  
• PCI DSS version 3.2.1 3.4  
• PCI DSS version 3.2.1 8.2.1 | • Encrypt data at rest | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/AWS-GR_ENCRYPTED_VOLUMES  
• US East (Ohio) arn:aws:controltower:us-east-2::control/AWS-GR_ENCRYPTED_VOLUMES  
• US West (Oregon) arn:aws:controltower:us-west-2::control/AWS-GR_ENCRYPTED_VOLUMES  
• Canada (Central) arn:aws:controltower:ca-central-1::control/AWS-GR_ENCRYPTED_VOLUMES  
• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AWS-GR_ENCRYPTED_VOLUMES  
• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AWS-GR_ENCRYPTED_VOLUMES  
• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/AWS-GR_ENCRYPTED_VOLUMES  
• Europe (Ireland) arn:aws:controltower:eu-west-1::control/AWS-GR_ENCRYPTED_VOLUMES  
• Europe (London) arn:aws:controltower:eu-west-2::control/AWS-GR_ENCRYPTED_VOLUMES  
• Europe (Stockholm) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>north-1::control/</td>
<td>north-1::control/</td>
<td>AWS-GR_ENCRYPTED_VOLUMES</td>
</tr>
<tr>
<td></td>
<td>aws-gr-encrypted_volumes</td>
<td>aws-gr-encrypted_volumes</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td>(Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/</td>
<td>aws-gr-encrypted_volumes</td>
</tr>
<tr>
<td></td>
<td>Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
<td>aws-gr-encrypted_volumes</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td>(Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/</td>
<td>aws-gr-encrypted_volumes</td>
</tr>
<tr>
<td></td>
<td>(Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/</td>
<td>aws-gr-encrypted_volumes</td>
</tr>
<tr>
<td>• South America</td>
<td>(São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/</td>
<td>aws-gr-encrypted_volumes</td>
</tr>
<tr>
<td>• US West (N.</td>
<td>California)</td>
<td>arn:aws:controltower:us-west-1::control/</td>
<td>aws-gr-encrypted_volumes</td>
</tr>
<tr>
<td></td>
<td>Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/</td>
<td>aws-gr-encrypted_volumes</td>
</tr>
<tr>
<td>• Middle East</td>
<td>(Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/</td>
<td>aws-gr-encrypted_volumes</td>
</tr>
<tr>
<td></td>
<td>(Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/</td>
<td>aws-gr-encrypted_volumes</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>AWS-GR_IAM_USER_MFA_ENABLED</td>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td>• Enforce least privilege</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/.AWS-GR_IAM_USER_MFA_ENABLED</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/.AWS-GR_IAM_USER_MFA_ENABLED</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 IA-2(1)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/.AWS-GR_IAM_USER_MFA_ENABLED</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 IA-2(2)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/.AWS-GR_IAM_USER_MFA_ENABLED</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 IA-2(6)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/.AWS-GR_IAM_USER_MFA_ENABLED</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.3.1</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.3.2</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.6</td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective, by Region</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>southeas-1::control/</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWS-GR_IAM_USER_MFA_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Frankfurt)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>central-1::control/</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWS-GR_IAM_USER_MFA_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Ireland)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>west-1::control/</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWS-GR_IAM_USER_MFA_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (London)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>west-2::control/</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWS-GR_IAM_USER_MFA_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Stockholm)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>north-1::control/</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWS-GR_IAM_USER_MFA_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>south-1::control/</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWS-GR_IAM_USER_MFA_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>northeas-2::control/</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWS-GR_IAM_USER_MFA_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>northeas-1::control/</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWS-GR_IAM_USER_MFA_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Paris)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>west-3::control/</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWS-GR_IAM_USER_MFA_ENABLED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>South America (São Paulo)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>east-1::control/</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_IAM_USER_MFA_ENABLED |           |                  | • **US West (N. California)**  
|                     |           |                  |  
|                     |           |                  |   arn:aws:controltower:us-west-1::control/  
|                     |           |                  |   AWS-GR_IAM_USER_MFA_ENABLED  
|                     |           |                  | • **Asia Pacific (Hong Kong)**  
|                     |           |                  |   arn:aws:controltower:ap-east-1::control/  
|                     |           |                  |   AWS-GR_IAM_USER_MFA_ENABLED  
|                     |           |                  | • **Asia Pacific (Jakarta)**  
|                     |           |                  |   arn:aws:controltower:ap-southeast-3::control/  
|                     |           |                  |   AWS-GR_IAM_USER_MFA_ENABLED  
|                     |           |                  | • **Asia Pacific (Osaka)**  
|                     |           |                  |   arn:aws:controltower:ap-northeast-3::control/  
|                     |           |                  |   AWS-GR_IAM_USER_MFA_ENABLED  
|                     |           |                  | • **Europe (Milan)**  
|                     |           |                  |   arn:aws:controltower:eu-south-1::control/  
|                     |           |                  |   AWS-GR_IAM_USER_MFA_ENABLED  
|                     |           |                  | • **Africa (Cape Town)**  
|                     |           |                  |   arn:aws:controltower:af-south-1::control/  
|                     |           |                  |   AWS-GR_IAM_USER_MFA_ENABLED  
|                     |           |                  | • **Middle East (Bahrain)**  
|                     |           |                  |   arn:aws:controltower:me-south-1::control/  
|                     |           |                  |   AWS-GR_IAM_USER_MFA_ENABLED  
|
## Control identifier, Framework, Control objective, Control API identifiers, by Region

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
<td>• NIST 800-53 Rev 5 AC-3&lt;br&gt;• NIST 800-53 Rev 5 AC-3(7)&lt;br&gt;• NIST 800-53 Rev 5 AC-4&lt;br&gt;• NIST 800-53 Rev 5 AC-4(21)&lt;br&gt;• NIST 800-53 Rev 5 AC-6&lt;br&gt;• NIST 800-53 Rev 5 SC-7&lt;br&gt;• NIST 800-53 Rev 5 SC-7(11)&lt;br&gt;• NIST 800-53 Rev 5 SC-7(16)&lt;br&gt;• NIST 800-53 Rev 5 SC-7(20)&lt;br&gt;• NIST 800-53 Rev 5 SC-7(21)&lt;br&gt;• NIST 800-53 Rev 5 SC-7(3)&lt;br&gt;• NIST 800-53 Rev 5 SC-7(4)&lt;br&gt;• NIST 800-53 Rev 5 SC-7(9)&lt;br&gt;• PCI DSS version 3.2.1 1.2.1&lt;br&gt;• PCI DSS version 3.2.1 1.3&lt;br&gt;• PCI DSS version 3.2.1 1.3.1&lt;br&gt;• PCI DSS version 3.2.1 1.3.2&lt;br&gt;• PCI DSS version 3.2.1 1.3.4&lt;br&gt;• PCI DSS version 3.2.1 2.2.2</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/&lt;br&gt;AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED&lt;br&gt;• US East (Ohio) arn:aws:controltower:us-east-2::control/&lt;br&gt;AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED&lt;br&gt;• US West (Oregon) arn:aws:controltower:us-west-2::control/&lt;br&gt;AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED&lt;br&gt;• Canada (Central) arn:aws:controltower:ca-central-1::control/&lt;br&gt;AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED&lt;br&gt;• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/&lt;br&gt;AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED&lt;br&gt;• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/&lt;br&gt;AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED&lt;br&gt;• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/&lt;br&gt;AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED&lt;br&gt;• Europe (Ireland) arn:aws:controltower:eu-west-1::control/&lt;br&gt;AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED&lt;br&gt;• Europe (London) arn:aws:controltower:eu-west-2::control/&lt;br&gt;AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
</tr>
</tbody>
</table>
| Europe (Milan)      |           |                  | arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>south-1::control/</td>
<td></td>
<td></td>
<td>AWS-GR_LAMBDA_FUNCTION_PUBLIC_ACCESS_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne)</td>
</tr>
</tbody>
</table>
### AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS</td>
<td>• CIS AWS Benchmark</td>
<td>• Enforce least privilege</td>
<td>• US East (N. Virginia) aws:controltower:us-east-1::control/</td>
</tr>
<tr>
<td></td>
<td>1.4 1.10</td>
<td></td>
<td>• US East (Ohio) aws:controltower:us-east-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) aws:controltower:us-west-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) aws:controltower:ca-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) aws:controltower:ap-southeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) aws:controltower:ap-southeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) aws:controltower:eu-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) aws:controltower:eu-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) aws:controltower:eu-west-2::control/</td>
</tr>
</tbody>
</table>

- NIST 800-53 Rev 5 AC-2(1)
- NIST 800-53 Rev 5 AC-3(15)
- NIST 800-53 Rev 5 IA-2(1)
- NIST 800-53 Rev 5 IA-2(2)
- NIST 800-53 Rev 5 IA-2(6)
- NIST 800-53 Rev 5 IA-2(8)
- PCI DSS version 3.2.1 2.2
- PCI DSS version 3.2.1 8.3.1
- PCI DSS version 3.2.1 8.3.2
- PCI DSS version 3.2.1 8.6

---

1557
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:saeast-1::control/AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:apeast-1::control/AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/AWS-GR_MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-</td>
</tr>
</tbody>
</table>
## AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW | • NIST 800-53 Rev 5 AC-4(21)  
• NIST 800-53 Rev 5 CM-7  
• NIST 800-53 Rev 5 SC-7  
• NIST 800-53 Rev 5 SC-7(11)  
• NIST 800-53 Rev 5 SC-7(16)  
• NIST 800-53 Rev 5 SC-7(21)  
• NIST 800-53 Rev 5 SC-7(4)  
• PCI DSS version 3.2.1 1.2.1  
• PCI DSS version 3.2.1 1.3  
• PCI DSS version 3.2.1 1.3.2 | • Limit network access | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1:control/  
AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2:control/  
AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW  
• US West (Oregon)  
AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW  
• Canada (Central)  
ar:n:aws:controltower:ca-central-1:control/  
AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW  
• Asia Pacific (Sydney)  
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td>AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>ARN:AWS:ControlTower:ap-southeast-1::control/</td>
<td></td>
<td>AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>ARN:AWS:ControlTower:eu-central-1::control/</td>
<td></td>
<td>AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>ARN:AWS:ControlTower:eu-west-1::control/</td>
<td></td>
<td>AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>ARN:AWS:ControlTower:eu-west-2::control/</td>
<td></td>
<td>AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>ARN:AWS:ControlTower:eu-north-1::control/</td>
<td></td>
<td>AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>ARN:AWS:ControlTower:ap-south-1::control/</td>
<td></td>
<td>AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>ARN:AWS:ControlTower:ap-northeast-2::control/</td>
<td></td>
<td>AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>ARN:AWS:ControlTower:ap-northeast-1::control/</td>
<td></td>
<td>AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>ARN:AWS:ControlTower:eu-west-3::control/</td>
<td></td>
<td>AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective, by Region</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>-----------------------------</td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-east-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:af-south-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:me-south-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:il-central-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-central-2::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:me-central-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-southeast-4::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/AWS-GR_NO_UNRESTRICTED_ROUTE_TO_IGW</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
<td></td>
</tr>
</tbody>
</table>
|                    | • Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/ AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK | • Asia Pacific (Jakarta) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>southeast-3::control/AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/AWS-GR_RDS_INSTANCE_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED | • NIST 800-53 Rev 5 AC-3  
• NIST 800-53 Rev 5 AC-3(7)  
• NIST 800-53 Rev 5 AC-4  
• NIST 800-53 Rev 5 AC-4(21)  
• NIST 800-53 Rev 5 AC-6  
• NIST 800-53 Rev 5 SC-7  
• NIST 800-53 Rev 5 SC-7(11)  
• NIST 800-53 Rev 5 SC-7(16)  
• NIST 800-53 Rev 5 SC-7(20)  
• NIST 800-53 Rev 5 SC-7(21)  
• NIST 800-53 Rev 5 SC-7(3)  
• NIST 800-53 Rev 5 SC-7(4)  
• NIST 800-53 Rev 5 SC-7(9)  
• PCI DSS version 3.2.1  
  1.2.1  
• PCI DSS version 3.2.1  
  1.3  
• PCI DSS version 3.2.1  
  1.3.1  
• PCI DSS version 3.2.1  
  1.3.2  
• PCI DSS version 3.2.1  
  1.3.4  
• PCI DSS version 3.2.1  
  1.3.6  
• PCI DSS version 3.2.1  
  2.2.2 | • Limit network access | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/  
AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/  
AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/  
AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED  
• Europe (Stockholm)  
arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>north-1::control/</td>
<td>AWS-</td>
<td></td>
<td>AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td>GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
<td>Mumbai</td>
<td>arn:aws:controltower:ap-south-1::control/AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td>GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
<td>Seoul</td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td>GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
<td>Tokyo</td>
<td>arn:aws:controltower:ap-northeast-1::control/AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
<td>Paris</td>
<td>arn:aws:controltower:eu-west-3::control/AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
</tr>
<tr>
<td>• South America</td>
<td>GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
<td>São Paulo</td>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
</tr>
<tr>
<td>• US West (N.</td>
<td>GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
<td>California</td>
<td>arn:aws:controltower:us-west-1::control/AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td>GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
<td>Hong Kong</td>
<td>arn:aws:controltower:ap-east-1::control/AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
</tr>
<tr>
<td>• Middle East</td>
<td>GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
<td>Bahrain</td>
<td>arn:aws:controltower:me-south-1::control/AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
</tr>
</tbody>
</table>
| • Asia Pacific     | GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED | Hyderabad | arn:aws:controltower:ap-
### AWS-GR_RDS_STORAGE_ENCRYPTED

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
</tr>
</thead>
</table>
| AWS-GR_RDS_STORAGE_ENCRYPTED | • CIS AWS Benchmark 2.3.1  
• NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-3(6)  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-28  
• NIST 800-53 Rev 5 SC-28(1)  
• NIST 800-53 Rev 5 SC-7(10)  
• NIST 800-53 Rev 5 SI-7(6)  
• PCI DSS version 3.2.1  
• PCI DSS version 3.2.1 8.2.1 |

### Control objective

- Encrypt data at rest

<table>
<thead>
<tr>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>south-2::control/AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED</td>
</tr>
</tbody>
</table>
| Middle East (UAE)  
arn:aws:controltower:me-central-1::control/AWS-GR_RDS_SNAPSHOTS_PUBLIC_PROHIBITED |

### AWS-GR_RDS_STORAGE_ENCRYPTED

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
</tr>
</thead>
</table>
| AWS-GR_RDS_STORAGE_ENCRYPTED | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/AWS-GR_RDS_STORAGE_ENCRYPTED  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/AWS-GR_RDS_STORAGE_ENCRYPTED  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/AWS-GR_RDS_STORAGE_ENCRYPTED  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED |

<table>
<thead>
<tr>
<th>Control objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypt data at rest</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/AWS-GR_RDS_STORAGE_ENCRYPTED  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/AWS-GR_RDS_STORAGE_ENCRYPTED  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/AWS-GR_RDS_STORAGE_ENCRYPTED  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-GR_RDS_STORAGE_ENCRYPTED</td>
<td>• Europe (Ireland)</td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td></td>
<td>• Europe (London)</td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td></td>
<td>• Europe (Stockholm)</td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td></td>
<td>• Europe (Paris)</td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo)</td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td></td>
<td>• US West (N. California)</td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/AWS-GR_RDS_STORAGE_ENCRYPTED</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Melbourne) |          |                   | arn:aws:controltower:ap-
### AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
<td>NIST 800-53 Rev 5</td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>1.2.1</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>1.3.1</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>1.3.2</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>1.2.1</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
</tr>
</tbody>
</table>
|                    | 1.3.1     |                  | Europe (Ireland) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>west-1::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/AWS-GR_REDSHIFT_CLUSTER_PUBLIC_ACCESS_CHECK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td>AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>south-1::control/</td>
<td></td>
<td>south-1::control/</td>
<td>AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America</td>
<td></td>
<td>South America</td>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>(São Paulo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N.</td>
<td></td>
<td>US West (N.</td>
<td>arn:aws:controltower:us-west-1::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td>California)</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific</td>
<td></td>
<td>Asia Pacific</td>
<td>arn:aws:controltower:ap-east-1::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific</td>
<td></td>
<td>Asia Pacific</td>
<td>arn:aws:controltower:ap-southeast-3::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>(Jakarta)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific</td>
<td></td>
<td>Asia Pacific</td>
<td>arn:aws:controltower:ap-northeast-3::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>(Osaka)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td><strong>Africa (Cape Town)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td><strong>Middle East (Bahrain)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td><strong>Israel (Tel Aviv)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td><strong>Europe (Zurich)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td><strong>Europe (Spain)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td><strong>Middle East (UAE)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td><strong>Asia Pacific (Melbourne)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/AWS-GR_REGION_DENY</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>AWS-GR_RESTRICTED_COMMON_PORTS</td>
<td>• CIS AWS Benchmark</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2(2)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-7</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
</tbody>
</table>
|                      | • NIST 800-53 Rev 5 SC-7(16)         |                   | • Europe (Stockholm) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>north-1::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/AWS-GR_RESTRICTED_COMMON_PORTS</td>
</tr>
</tbody>
</table>
|                    |           |                   | • Asia Pacific (Hyderabad) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-GR_RESTRICTED_SSH</td>
<td>CIS AWS Benchmark 1.4 5.2</td>
<td>Limit network access</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AWS-GR_RESTRICTED_SSH

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-GR_RESTRICTED_SSH</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ AWS-GR_RESTRICTED_SSH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ AWS-GR_RESTRICTED_SSH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ AWS-GR_RESTRICTED_SSH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ AWS-GR_RESTRICTED_SSH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ AWS-GR_RESTRICTED_SSH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ AWS-GR_RESTRICTED_SSH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ AWS-GR_RESTRICTED_SSH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ AWS-GR_RESTRICTED_SSH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>AWS-GR_RESTRICTED_SSH</td>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/</td>
<td>AWS-GR_RESTRICTED_SSH</td>
</tr>
<tr>
<td></td>
<td>EUR (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/</td>
<td>AWS-GR_RESTRICTED_SSH</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/</td>
<td>AWS-GR_RESTRICTED_SSH</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
<td>AWS-GR_RESTRICTED_SSH</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/</td>
<td>AWS-GR_RESTRICTED_SSH</td>
</tr>
<tr>
<td></td>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/</td>
<td>AWS-GR_RESTRICTED_SSH</td>
</tr>
<tr>
<td></td>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/</td>
<td>AWS-GR_RESTRICTED_SSH</td>
</tr>
<tr>
<td></td>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/</td>
<td>AWS-GR_RESTRICTED_SSH</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/</td>
<td>AWS-GR_RESTRICTED_SSH</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_RESTRICT_ROOT_USER | • NIST 800-53 Rev 5 AC-2(1)  
• NIST 800-53 Rev 5 AC-3(15)  
• NIST 800-53 Rev 5 AC-3(7)  
• NIST 800-53 Rev 5 AC-6  
• NIST 800-53 Rev 5 AC-6(10)  
• NIST 800-53 Rev 5 AC-6(2)  
• PCI DSS version 3.2.1  
• PCI DSS version 3.2.1  
• PCI DSS version 3.2.1  
• PCI DSS version 3.2.1  
• PCI DSS version 3.2.1  
• PCI DSS version 3.2.1  | • Enforce least privilege | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
AWS-GR_RESTRICT_ROOT_USER  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
AWS-GR_RESTRICT_ROOT_USER  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
AWS-GR_RESTRICT_ROOT_USER  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
AWS-GR_RESTRICT_ROOT_USER  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
AWS-GR_RESTRICT_ROOT_USER  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
AWS-GR_RESTRICT_ROOT_USER  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/  
AWS-GR_RESTRICT_ROOT_USER |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Asia Pacific (Hong Kong) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>AWS-GR_RESTRICT_ROOT_USER</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/ AWS-GR_RESTRICT_ROOT_USER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
<td>• CIS AWS Benchmark ACCESS_KEYS</td>
<td>• Enforce least privilege</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6(10)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/</td>
<td></td>
<td>AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEY</td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/</td>
<td></td>
<td>AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEY</td>
</tr>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/</td>
<td></td>
<td>AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEY</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/</td>
<td></td>
<td>AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEY</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/</td>
<td></td>
<td>AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEY</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
<td></td>
<td>AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEY</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/</td>
<td></td>
<td>AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEY</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/</td>
<td></td>
<td>AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEY</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/</td>
<td></td>
<td>AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEY</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/</td>
<td></td>
<td>AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEY</td>
</tr>
</tbody>
</table>

1584
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/ AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/ AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/ AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/ AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/ AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/ AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/ AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/ AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/ AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/ AWS-GR_RESTRICT_ROOT_USER_ACCESS_KEYS</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>----------------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION | • NIST 800-53 Rev 5 CM-2(2)  
• NIST 800-53 Rev 5 CM-3  
• NIST 800-53 Rev 5 CM-3(8)  
• NIST 800-53 Rev 5 SA-8(19) | • Improve resiliency          | • US East (N. Virginia)  
ar:aws:controltower:us-east-1::control/  
AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION  
• US East (Ohio)  
ar:aws:controltower:us-east-2::control/  
AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION  
• US West (Oregon)  
ar:aws:controltower:us-west-2::control/  
AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION  
• Canada (Central)  
ar:aws:controltower:ca-central-1::control/  
AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION  
• Asia Pacific (Sydney)  
ar:aws:controltower:ap-southeast-2::control/  
AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Control objective, by Region</strong></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>AWS-GR_RESTRICT_S3_CROSS_REGION_REPLICATION</td>
<td>US West (N. California)</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td>Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td>Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td>Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td>Israel (Tel Aviv)</td>
</tr>
</tbody>
</table>
|                  | Europe (Zurich) | arn:aws:controltower:eu-central-2::control/ AWS-GR_RESTRICT_S3_CROSS_REGION_Repli...
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/</td>
<td>AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/</td>
<td>AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA</td>
<td></td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/</td>
<td>AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA</td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/</td>
<td>AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA</td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/</td>
<td>AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA</td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/</td>
<td>AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/</td>
<td>AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
<td>AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/</td>
<td>AWS-GR_RESTRICT_S3_DELETE_WITHOUT_MFA</td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>AWS-GR.Restrict_S3_DELETE_WITHOUT_MFA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>South America</strong> (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR.Restrict_S3_DELETE_WITHOUT_MFA</td>
</tr>
<tr>
<td><strong>US West (N. California)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/AWS-GR.Restrict_S3_DELETE_WITHOUT_MFA</td>
</tr>
<tr>
<td><strong>Asia Pacific</strong> (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/AWS-GR.Restrict_S3_DELETE_WITHOUT_MFA</td>
</tr>
<tr>
<td><strong>Asia Pacific (Jakarta)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/AWS-GR.Restrict_S3_DELETE_WITHOUT_MFA</td>
</tr>
<tr>
<td><strong>Asia Pacific (Osaka)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/AWS-GR.Restrict_S3_DELETE_WITHOUT_MFA</td>
</tr>
<tr>
<td><strong>Europe (Milan)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/AWS-GR.Restrict_S3_DELETE_WITHOUT_MFA</td>
</tr>
<tr>
<td><strong>Africa (Cape Town)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/AWS-GR.Restrict_S3_DELETE_WITHOUT_MFA</td>
</tr>
<tr>
<td><strong>Middle East (Bahrain)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/AWS-GR.Restrict_S3_DELETE_WITHOUT_MFA</td>
</tr>
<tr>
<td><strong>Israel (Tel Aviv)</strong></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/AWS-GR.Restrict_S3_DELETE_WITHOUT_MFA</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_ROOT_ACCOUNT_MFA_ENABLED | • CIS AWS Benchmark ENABLED  
  • NIST 800-53 Rev 5 AC-2(1)  
  • NIST 800-53 Rev 5 AC-3(15)  
  • NIST 800-53 Rev 5 IA-2(1)  
  • NIST 800-53 Rev 5 IA-2(2)  
  • NIST 800-53 Rev 5 IA-2(6)  
  • NIST 800-53 Rev 5 IA-2(8)  
  • PCI DSS version 3.2.1 2.1  
  • Enforce least privilege | • US East (N. Virginia)  
  arn:aws:controltower:us-east-1::control/  
  AWS-GR_ROOT_ACCOUNT_MFA_ENABLED  
  • US East (Ohio)  
  arn:aws:controltower:us-east-2::control/  
  AWS-GR_ROOT_ACCOUNT_MFA_ENABLED  
  • US West (Oregon)  
  arn:aws:controltower:us-west-2::control/  
  AWS-GR_ROOT_ACCOUNT_MFA_ENABLED |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/AWS-GR_ROOT_ACCOUNT_MFA_ENABLED</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.3.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AWS-GR_ROOT_ACCOUNT_MFA_ENABLED</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.3.2</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/AWS-GR_ROOT_ACCOUNT_MFA_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/AWS-GR_ROOT_ACCOUNT_MFA_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/AWS-GR_ROOT_ACCOUNT_MFA_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/AWS-GR_ROOT_ACCOUNT_MFA_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/AWS-GR_ROOT_ACCOUNT_MFA_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/AWS-GR_ROOT_ACCOUNT_MFA_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/AWS-GR_ROOT_ACCOUNT_MFA_ENABLED</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>AWS-GR_ROOT_ACCOUNT_MFA_ENABLED</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_ROOT_ACCOUNT_MFA_ENABLED.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_ROOT_ACCOUNT_MFA_ENABLED.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_ROOT_ACCOUNT_MFA_ENABLED.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_ROOT_ACCOUNT_MFA_ENABLED.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_ROOT_ACCOUNT_MFA_ENABLED.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_ROOT_ACCOUNT_MFA_ENABLED.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Milan) arn:aws:controltower:eu-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_ROOT_ACCOUNT_MFA_ENABLED.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Africa (Cape Town) arn:aws:controltower:af-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_ROOT_ACCOUNT_MFA_ENABLED.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Middle East (Bahrain) arn:aws:controltower:me-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_ROOT_ACCOUNT_MFA_ENABLED.</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-----------------------------------------------</td>
<td>------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
<td>• CIS AWS Benchmark</td>
<td>• Enforce least privilege</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-21</td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(3)</td>
<td>Asia Pacific (Singapore)</td>
<td><code>arn:aws:controltower:ap-southeast-1::control/AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</code></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td>Europe (Frankfurt)</td>
<td><code>arn:aws:controltower:eu-central-1::control/AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</code></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(9)</td>
<td>Europe (Ireland)</td>
<td><code>arn:aws:controltower:eu-west-1::control/AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</code></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.2.1</td>
<td>Europe (London)</td>
<td><code>arn:aws:controltower:eu-west-2::control/AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</code></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3</td>
<td>Europe (Stockholm)</td>
<td><code>arn:aws:controltower:eu-north-1::control/AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</code></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.1</td>
<td>Asia Pacific (Mumbai)</td>
<td><code>arn:aws:controltower:ap-south-1::control/AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</code></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.2</td>
<td>Asia Pacific (Seoul)</td>
<td><code>arn:aws:controltower:ap-northeast-2::control/AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</code></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.4</td>
<td>Asia Pacific (Tokyo)</td>
<td><code>arn:aws:controltower:ap-northeast-1::control/AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</code></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.6</td>
<td>Europe (Paris)</td>
<td><code>arn:aws:controltower:eu-west-3::control/AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</code></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2</td>
<td>South America (São Paulo)</td>
<td><code>arn:aws:controltower:sa-east-1::control/AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</code></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>US West (N. California)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Hong Kong)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Jakarta)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Osaka)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Europe (Milan)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Africa (Cape Town)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Middle East (Bahrain)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Middle East (UAE)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_S3_ACCOUNT_LEVEL_PUBLIC_ACCESS_BLOCKS_PERIODIC</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED | • CIS AWS Benchmark<br>AC-21<br>NIST 800-53 Rev 5 AC-3<br>NIST 800-53 Rev 5 AC-3(7)<br>NIST 800-53 Rev 5 AC-4<br>NIST 800-53 Rev 5 AC-4(21)<br>NIST 800-53 Rev 5 AC-6<br>NIST 800-53 Rev 5 SC-7<br>NIST 800-53 Rev 5 SC-7(11)<br>NIST 800-53 Rev 5 SC-7(16)<br>NIST 800-53 Rev 5 SC-7(20)<br>NIST 800-53 Rev 5 SC-7(21)<br>NIST 800-53 Rev 5 SC-7(3)<br>NIST 800-53 Rev 5 SC-7(4)<br>NIST 800-53 Rev 5 SC-7(9)<br>PCI DSS version 3.2.1 1.2.1<br>PCI DSS version 3.2.1 1.3<br>PCI DSS version 3.2.1 1.3.1<br>PCI DSS version 3.2.1 1.3.2<br>PCI DSS version 3.2.1 1.3.4<br>PCI DSS version 3.2.1 1.3.6<br>PCI DSS version 3.2.1 2.2 | • Enforce least privilege | • US East (N. Virginia)<br>arn:aws:controltower:us-east-1::control/<br>AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED<br>• US East (Ohio)<br>arn:aws:controltower:us-east-2::control/<br>AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED<br>• US West (Oregon)<br>arn:aws:controltower:us-west-2::control/<br>AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED<br>• Canada (Central)<br>arn:aws:controltower:ca-central-1::control/<br>AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED<br>• Asia Pacific (Sydney)<br>arn:aws:controltower:ap-southeast-2::control/<br>AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED<br>• Asia Pacific (Singapore)<br>arn:aws:controltower:ap-southeast-1::control/<br>AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED<br>• Europe (Frankfurt)<br>arn:aws:controltower:eu-central-1::control/<br>AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED<br>• Europe (Ireland)<br>arn:aws:controltower:eu-west-1::control/<br>AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED<br>• Europe (London)<br>arn:aws:controltower:eu-west-2::control/<br>AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED<br>• Europe (Stockholm)<br>arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td>AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
<td>north-1::control/ AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Milan)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Africa (Cape Town)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Middle East (Bahrain)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Israel (Tel Aviv)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Zurich)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Spain)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Asia Pacific (Hyderabad)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Middle East (UAE)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Asia Pacific (Melbourne)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/AWS-GR_S3_BUCKET_PUBLIC_READ_PROHIBITED</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED | - CIS AWS Benchmark | • Enforce least privilege | - US East (N. Virginia)  
|                     | - NIST 800-53 Rev 5 |  
|                     | AC-21 |  
|                     | - NIST 800-53 Rev 5 |  
|                     | AC-3 |  
|                     | - NIST 800-53 Rev 5 |  
|                     | AC-3(7) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | AC-4 |  
|                     | - NIST 800-53 Rev 5 |  
|                     | AC-4(21) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | AC-6 |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7 |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(11) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(16) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(20) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(21) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(3) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(4) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(9) |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 1.2.1 |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 1.3 |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 1.3.1 |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 1.3.2 |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 1.3.4 |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 1.3.6 |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 2.2 |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7 |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(11) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(16) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(20) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(21) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(3) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(4) |  
|                     | - NIST 800-53 Rev 5 |  
|                     | SC-7(9) |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 1.2.1 |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 1.3 |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 1.3.1 |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 1.3.2 |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 1.3.4 |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 1.3.6 |  
|                     | - PCI DSS version 3.2.1 |  
|                     | 2.2 |  

- US East (Ohio)  
us-east-2::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED  
- US West (Oregon)  
us-west-2::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED  
- Canada (Central)  
ca-central-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED  
- Asia Pacific (Sydney)  
ap-southeast-2::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED  
- Asia Pacific (Singapore)  
ap-southeast-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED  
- Europe (Frankfurt)  
eu-central-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED  
- Europe (Ireland)  
eu-west-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED  
- Europe (London)  
eu-west-2::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED  
- Europe (Stockholm)  
eu-1601
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>north-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Milan)</strong> arn:aws:controltower:eu-south-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Africa (Cape Town)</strong> arn:aws:controltower:af-south-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Middle East (Bahrain)</strong> arn:aws:controltower:me-south-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Israel (Tel Aviv)</strong> arn:aws:controltower:il-central-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Zurich)</strong> arn:aws:controltower:eu-central-2::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Spain)</strong> arn:aws:controltower:eu-south-2::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Asia Pacific (Hyderabad)</strong> arn:aws:controltower:ap-south-2::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Middle East (UAE)</strong> arn:aws:controltower:me-central-1::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Asia Pacific (Melbourne)</strong> arn:aws:controltower:ap-southeast-4::control/AWS-GR_S3_BUCKET_PUBLIC_WRITE_PROHIBITED</td>
</tr>
</tbody>
</table>


### AWS-GR_S3_VERSIONING_ENABLED

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| AWS-GR_S3_VERSIONING_ENABLED | • CIS AWS Benchmark 2.1.3  
• NIST 800-53 Rev 5 AU-9(2)  
• NIST 800-53 Rev 5 CP-10  
• NIST 800-53 Rev 5 CP-6  
• NIST 800-53 Rev 5 CP-6(1)  
• NIST 800-53 Rev 5 CP-6(2)  
• NIST 800-53 Rev 5 CP-9  
• NIST 800-53 Rev 5 SC-5(2)  
• NIST 800-53 Rev 5 SI-12  
• NIST 800-53 Rev 5 SI-13(5)  
• PCI DSS version 3.2.1 10.5.5 | • Optimize costs  
• Improve availability | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/AWS-GR_S3_VERSIONING_ENABLED  
• US East (Ohio) arn:aws:controltower:us-east-2::control/AWS-GR_S3_VERSIONING_ENABLED  
• US West (Oregon) arn:aws:controltower:us-west-2::control/AWS-GR_S3_VERSIONING_ENABLED  
• Canada (Central) arn:aws:controltower:ca-central-1::control/AWS-GR_S3_VERSIONING_ENABLED  
• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AWS-GR_S3_VERSIONING_ENABLED  
• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AWS-GR_S3_VERSIONING_ENABLED  
• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/AWS-GR_S3_VERSIONING_ENABLED  
• Europe (Ireland) arn:aws:controltower:eu-west-1::control/AWS-GR_S3_VERSIONING_ENABLED  
• Europe (London) arn:aws:controltower:eu-west-2::control/AWS-GR_S3_VERSIONING_ENABLED  
• Europe (Stockholm) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>north-1::control/</td>
<td></td>
<td></td>
<td>aws-gr_s3_versioning_enabled</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/aws-gr_s3_versioning_enabled</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/aws-gr_s3_versioning_enabled</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/aws-gr_s3_versioning_enabled</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/aws-gr_s3_versioning_enabled</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/aws-gr_s3_versioning_enabled</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/aws-gr_s3_versioning_enabled</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/aws-gr_s3_versioning_enabled</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/aws-gr_s3_versioning_enabled</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/aws-gr_s3_versioning_enabled</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_S3_VERSIONING_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Milan)</td>
<td>• arn:aws:controltower:eu-south-1::control/AWS-GR_S3_VERSIONING_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Africa (Cape Town)</td>
<td>• arn:aws:controltower:af-south-1::control/AWS-GR_S3_VERSIONING_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Middle East (Bahrain)</td>
<td>• arn:aws:controltower:me-south-1::control/AWS-GR_S3_VERSIONING_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Israel (Tel Aviv)</td>
<td>• arn:aws:controltower:il-central-1::control/AWS-GR_S3_VERSIONING_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Zurich)</td>
<td>• arn:aws:controltower:eu-central-2::control/AWS-GR_S3_VERSIONING_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Spain)</td>
<td>• arn:aws:controltower:eu-south-2::control/AWS-GR_S3_VERSIONING_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia Pacific (Hyderabad)</td>
<td>• arn:aws:controltower:ap-south-2::control/AWS-GR_S3_VERSIONING_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Middle East (UAE)</td>
<td>• arn:aws:controltower:me-central-1::control/AWS-GR_S3_VERSIONING_ENABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia Pacific (Melbourne)</td>
<td>• arn:aws:controltower:ap-southeast-4::control/AWS-GR_S3_VERSIONING_ENABLED</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>------------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS | • NIST 800-53 Rev 5 AC-3  
• NIST 800-53 Rev 5 AC-3(7)  
• NIST 800-53 Rev 5 AC-4  
• NIST 800-53 Rev 5 AC-4(21)  
• NIST 800-53 Rev 5 AC-6  
• NIST 800-53 Rev 5 SC-7  
• NIST 800-53 Rev 5 SC-7(11)  
• NIST 800-53 Rev 5 SC-7(16)  
• NIST 800-53 Rev 5 SC-7(20)  
• NIST 800-53 Rev 5 SC-7(21)  
• NIST 800-53 Rev 5 SC-7(3)  
• NIST 800-53 Rev 5 SC-7(4)  
• NIST 800-53 Rev 5 SC-7(9)  
• PCI DSS version 3.2.1  
  1.2.1  
• PCI DSS version 3.2.1  
  1.3  
• PCI DSS version 3.2.1  
  1.3.1  
• PCI DSS version 3.2.1  
  1.3.2  
• PCI DSS version 3.2.1  
  1.3.4  
• PCI DSS version 3.2.1  
  1.3.6  
• PCI DSS version 3.2.1  
  2.2.2 | • Limit network access | • US East (N. Virginia)  
ar:n:arn:aws:controltower:us-east-1::control/  
AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS  
• US East (Ohio)  
ar:n:arn:aws:controltower:us-east-2::control/  
AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS  
• US West (Oregon)  
ar:n:arn:aws:controltower:us-west-2::control/  
AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS  
• Canada (Central)  
ar:n:arn:aws:controltower:ca-central-1::control/  
AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS  
• Asia Pacific (Sydney)  
ar:n:arn:aws:controltower:ap-southeast-2::control/  
AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS  
• Asia Pacific  
(Singapore)  
ar:n:arn:aws:controltower:ap-southeast-1::control/  
AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS  
• Europe (Frankfurt)  
ar:n:arn:aws:controltower:eu-central-1::control/  
AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS  
• Europe (Ireland)  
ar:n:arn:aws:controltower:eu-west-1::control/  
AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS  
• Europe (London)  
ar:n:arn:aws:controltower:eu-west-2::control/  
AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/ AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/ AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/ AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/ AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/ AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/ AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/ AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/ AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/ AWS-GR_SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| AWS-GR_SSM_DOCUMENT_NOT_PUBLIC | • NIST 800-53 Rev 5 AC-3  
• NIST 800-53 Rev 5 AC-3(7)  
• NIST 800-53 Rev 5 AC-4  
• NIST 800-53 Rev 5 AC-4(21)  
• NIST 800-53 Rev 5 AC-6  
• NIST 800-53 Rev 5 SC-7  
• NIST 800-53 Rev 5 SC-7(11)  
• NIST 800-53 Rev 5 SC-7(16)  
• NIST 800-53 Rev 5 SC-7(20)  
• NIST 800-53 Rev 5 SC-7(21)  
• NIST 800-53 Rev 5 SC-7(3)  
• NIST 800-53 Rev 5 SC-7(4)  
• NIST 800-53 Rev 5 SC-7(9)  
• PCI DSS version 3.2.1 1.2.1  
• PCI DSS version 3.2.1 1.3  
• PCI DSS version 3.2.1 1.3.1  
• PCI DSS version 3.2.1 1.3.2  
• PCI DSS version 3.2.1 1.3.4  
• PCI DSS version 3.2.1 2.2.2 | • Limit network access | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/aws-GR_SSM_DOCUMENT_NOT_PUBLIC  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/aws-GR_SSM_DOCUMENT_NOT_PUBLIC  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/aws-GR_SSM_DOCUMENT_NOT_PUBLIC  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/aws-GR_SSM_DOCUMENT_NOT_PUBLIC  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/aws-GR_SSM_DOCUMENT_NOT_PUBLIC  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/aws-GR_SSM_DOCUMENT_NOT_PUBLIC  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/aws-GR_SSM_DOCUMENT_NOT_PUBLIC  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/aws-GR_SSM_DOCUMENT_NOT_PUBLIC  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/aws-GR_SSM_DOCUMENT_NOT_PUBLIC  
• Europe (Stockholm)  
arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>north-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_SSM_DOCUMENT_NOT_PUBLIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_SSM_DOCUMENT_NOT_PUBLIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_SSM_DOCUMENT_NOT_PUBLIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_SSM_DOCUMENT_NOT_PUBLIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_SSM_DOCUMENT_NOT_PUBLIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_SSM_DOCUMENT_NOT_PUBLIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_SSM_DOCUMENT_NOT_PUBLIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_SSM_DOCUMENT_NOT_PUBLIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_SSM_DOCUMENT_NOT_PUBLIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
<td>• NIST 800-53 Rev 5</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
</tbody>
</table>

- **Europe (Milan)**
  arn:aws:controltower:eusouth-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

- **Africa (Cape Town)**
  arn:aws:controltower:afsouth-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

- **Middle East (Bahrain)**
  arn:aws:controltower:mesouth-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

- **Europe (Zurich)**
  arn:aws:controltower:eucentral-2::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

- **Europe (Spain)**
  arn:aws:controltower:eusouth-2::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

- **Asia Pacific (Hyderabad)**
  arn:aws:controltower:apsouth-2::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

- **Middle East (UAE)**
  arn:aws:controltower:me-central-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

**AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED**

- **Europe (Milan)**
  arn:aws:controltower:eusouth-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

- **Africa (Cape Town)**
  arn:aws:controltower:afsouth-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

- **Middle East (Bahrain)**
  arn:aws:controltower:mesouth-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

- **Europe (Zurich)**
  arn:aws:controltower:eucentral-2::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

- **Europe (Spain)**
  arn:aws:controltower:eusouth-2::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

- **Asia Pacific (Hyderabad)**
  arn:aws:controltower:apsouth-2::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED

- **Middle East (UAE)**
  arn:aws:controltower:me-central-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td></td>
<td>east-1::control/AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/ AWS-GR_SUBNET_AUTO_ASSIGN_PUBLIC_IP_DISABLED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.ACM.PR.1        | None      | • Protect configurations | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
CWIZNCR1IXHMW  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
SOEEZVLQQMBP |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Canada (Central)</td>
<td>arn:aws:controltower:ca-central-1::control/CKCKTGLVEKYB</td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/OTJFGEEQXKHE</td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/JHLOZPXDFZLZ</td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/EFGFCKDLGVOR</td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/RMPGJNTOZYED</td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/TMHLCRNZFYCN</td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/XJPCVFFVNVNLH</td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/YSLZGQSFAAGK</td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ZYLCIKKLJKLN</td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/XFRRFFLMUBDQ</td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/DEOJHGUDMVVV</td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/</td>
<td></td>
<td>MHGHBXOCHYCS</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/</td>
<td></td>
<td>OGGUGDZQUZYR</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/</td>
<td></td>
<td>DIIUHEHAUSSPQ</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
<td></td>
<td>IYXOFXBXLMGS</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
<td></td>
<td>PFBKTWXMIMIV</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/</td>
<td></td>
<td>AJYPIRPBHBRU</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/</td>
<td></td>
<td>ZELXBEFAIZK</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/</td>
<td></td>
<td>MJAMNXHSTEMV</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/</td>
<td></td>
<td>AIMLPTHSIIAM</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/</td>
<td></td>
<td>JSJTOIBKUZXK</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/</td>
<td></td>
<td>CDFRPKVAEEBA</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Hyderabad) | arn:aws:controltower:ap-
### CT.APIGATEWAY.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.APIGATEWAY.PR.1</td>
<td>• NIST 800-53 Rev 5 AC-4(26)</td>
<td>• Establish logging and monitoring</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/THVNUTMFZVUS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/XRIIZCMYLTPU</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/SKIBWKYUQAAC</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/EUYZNAPVCUUS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/MECUGHLRJHG</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/QFZCLBSXX8KM</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/XHEVQAZTSEA2</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-central-1::control/XHEVQAZTSEA2</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td>• PCI DSS version 3.2.1 10.2.1</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td>• PCI DSS version 3.2.1 10.2.7</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td></td>
<td>• PCI DSS version 3.2.1 10.3.1</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.2.7</td>
<td></td>
<td>• PCI DSS version 3.2.1 10.3.1</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td></td>
<td></td>
<td>west-1::control/KBXFFJCCXCCZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ VLFAZTXPNOXB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ RPLZIVLHUVE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ OWNNOAXIDVKF</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ ASAMDCSLJXZV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ ETKWXIYCRYUF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/ VJLLSIZIUQRW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/ XGXELVOLINOS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/ ASCJCHWUCXDE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/ HQJNVX0QEKEX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/ CBCSCTURQXDP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/ SQJBDNVUH0VM</td>
</tr>
</tbody>
</table>
## CT.APIGATEWAY.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.APIGATEWAY.PR.2</td>
<td>NIST 800-53 Rev 5 CA-7, PCI DSS version 3.2.1 10.1</td>
<td>Establish logging and monitoring</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/EHSOKSSMVFVF</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td>• US East (Ohio)</td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/MQXZJKMXOREU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td>• US West (Oregon)</td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/XAZHJTQ8XMLM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.7</td>
<td>• Canada (Central)</td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/RPOXLFOSAVOI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/EAPHSJQRHZUB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/IFWWBITJQWQ0</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td>• Europe (Frankfurt)</td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/PEIQUNSHAAIJC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td>• Europe (Ireland)</td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/WEFXXKWQELZP</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td>• Europe (London)</td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/EDQVH50AJ0DX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td>• Europe (Stockholm)</td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/J2XXPWRFZLER</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/YVJSVPYRBSBH</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/MBGEYEHC0BBU</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/XSDBGMLIVBR</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:eu-west-3::control/GUYFEHMSFWNE</code></td>
</tr>
<tr>
<td>• South America</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:sa-east-1::control/ MQDMDGQZLLEE</code></td>
</tr>
<tr>
<td>(São Paulo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:us-west-1::control/ WITVQTILCFXZ</code></td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-east-1::control/ KAUPNLFAXLZW</code></td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-southeast-3::control/ EGXWMKYDFRSK</code></td>
</tr>
<tr>
<td>(Jakarta)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-northeast-3::control/ SXPSZGLUEEZH</code></td>
</tr>
<tr>
<td>(Osaka)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:eu-south-1::control/ MWDNBKQDCNCE</code></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:af-south-1::control/ YCKXUGPJLVQ</code></td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Bahrain)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:me-south-1::control/ BMPTRRAHQMJC</code></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:il-central-1::control/ BZHXNKHANLSRP</code></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:eu-central-2::control/ MRKDCBADGMLT</code></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:eu-central-3::control/ WITVQTILCFXZ</code></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>CT.APIGATEWAY.PR.3</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/FIMWSVKNFANN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/DZDLWPOKGTPI</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/XKODHWTZTZ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/NGD1EPXBGZNX</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ESHXZMSBZXKE</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>central-1::control/TAIPGHUJACMN</td>
<td>Europe (Ireland)</td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/DDAUQMQWAFBZ</td>
</tr>
<tr>
<td>Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/BWHEHVVZVYOL</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:aws:controltower:eu-north-1::control/MQGSDYXVDQAD</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/HEBRXKGJ0RFL</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/NBLSRDRBDJX0</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/FGGPHPWIRFWB</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/QSKRUWXUNEUH</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/EISQVHJ3JDWH</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/KSZKAGOPPWPH</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/NLXAPQRPKIYU</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/TRNOXUPEBWO0</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/CDPDATWIGGUH</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/KCSNFBWRTAXT</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/ILKACTDDSJMT</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/LJZPLSKCPFIT</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/PEZSUQGJPLXV</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/BRRZGXLVBEXS</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/MYDVOFUAZSBV</td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/DNUEOQSFWTRD</td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/PTRXVPJHHFKP</td>
</tr>
<tr>
<td>• Asia Pacific (Melbourne)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/JSVPQLCSVWLG</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.APIGATEWAY.PR.4 | • NIST 800-53 Rev 5 AC-4(26)  
• NIST 800-53 Rev 5 AU-10  
• NIST 800-53 Rev 5 AU-12  
• NIST 800-53 Rev 5 AU-2  
• NIST 800-53 Rev 5 AU-3  
• NIST 800-53 Rev 5 AU-6(3)  
• NIST 800-53 Rev 5 AU-6(4)  
• NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 SC-7(9)  
• NIST 800-53 Rev 5 SI-7(8)  
• PCI DSS version 3.2.1 10.1  
• PCI DSS version 3.2.1 10.2.1  
• PCI DSS version 3.2.1 10.2.4  
• PCI DSS version 3.2.1 10.2.7  
• PCI DSS version 3.2.1 10.3.1  
• PCI DSS version 3.2.1 10.3.2  
• PCI DSS version 3.2.1 10.3.3  
• PCI DSS version 3.2.1 10.3.4  
• PCI DSS version 3.2.1 10.3.5  
• PCI DSS version 3.2.1 10.3.6 | • Establish logging and monitoring | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/CSLQUIKVRGVG  
• US East (Ohio) arn:aws:controltower:us-east-2::control/ZZRQCTUQT8PU  
• US West (Oregon) arn:aws:controltower:us-west-2::control/CQKSRVXTWTEC  
• Canada (Central) arn:aws:controltower:ca-central-1::control/WDEOZRYGKEQK  
• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/WUQQUVNLXQDU  
• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/KHGSJQSDYFLH  
• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/HWAVU0ESYCQT  
• Europe (Ireland) arn:aws:controltower:eu-west-1::control/VEC533J8MGM  
• Europe (London) arn:aws:controltower:eu-west-2::control/FADFWQHALJXO  
• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/HQMBNTKLT0G  
• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/YRNSAX0GSJ3Y
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/MULDPL0TLRAJ</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/QSCLQFQFNJQXE</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/OOIWAZTWKKPE</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/YNMPINWZYZY</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/ITPBHOOWRJGI</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/KNVAYHZACOEU</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/BOOKWYPOXMTQ</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/FAGHQEVLPPXR</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/IYMXH1OFYD</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/LODJMM5VEOMN</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/KC1FTRLVLAAAP</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.APIGATEWAY.PR.5 | • NIST 800-53 Rev 5 AC-3<br>• NIST 800-53 Rev 5 CM-2<br>• NIST 800-53 Rev 5 CM-2(2)<br>• PCI DSS version 3.2.1 7.1.1<br>• PCI DSS version 3.2.1 7.2.1<br>• PCI DSS version 3.2.1 7.2.2 | • Use strong authentication | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/ YNAHJMRFSATY<br>• US East (Ohio) arn:aws:controltower:us-east-2::control/ KRLMXVLLTUAN<br>• US West (Oregon) arn:aws:controltower:us-west-2::control/ QSEYQKHJZQ0<br>• Canada (Central) arn:aws:controltower:ca-central-1::control/ NMHBYPLKKHGTX<br>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ GDMZARUABRDK
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>southeast-2::control/RBLHXKSFLKZA</td>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/FXZXEDHLSFKXJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/UQCOJSUFRRVSB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/AKVOXJCRARVM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/YYOCBPGNDCOQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/BVSVXEMRBTNG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/UXJBMYYDQHMX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/RFIFJOHVLBABB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/UKYEADDRMENU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/JNGATIIVCCY5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/ZHWWUFJCEAJBDB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/YHRHPQEAQQFX</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective, by Region</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------------------</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/EGRYMA3BFUD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/PAPTPLXSJUFL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/GVUDXU0KDLN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/BROASONNAOUZH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/WSNSTAYGKFCU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/HLDJVPJSDPEY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/EFKJPLZIUAII</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/WHSNERTF3EH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/ATODWGMZUCX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/E0XAVTBQHMIE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/ILMJIBFINEK</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Asia Pacific (Melbourne) | arn:aws:controltower:ap-
### Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
CT.APPSYNC.PR.1 | • NIST 800-53 Rev 5 AC-4(26) | • Establish logging and monitoring | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/ECXYQKUETWKC
• NIST 800-53 Rev 5 AU-10 | | | • US East (Ohio) arn:aws:controltower:us-east-2::control/BCZEGAHZH5II
• NIST 800-53 Rev 5 AU-12 | | | • US West (Oregon) arn:aws:controltower:us-west-2::control/3JBUYRGFM2AE
• NIST 800-53 Rev 5 AU-2 | | | • Canada (Central) arn:aws:controltower:canada-central-1::control/KRLWEXIIRJPA
• NIST 800-53 Rev 5 AU-3 | | | • Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/VZIWINLZXVMA
• NIST 800-53 Rev 5 AU-6(3) | | | • Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/SZSRSLXO0XB
• NIST 800-53 Rev 5 AU-6(4) | | | • Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/MTIRJIPWCKMX
• NIST 800-53 Rev 5 CA-7 | | | • Europe (Ireland) arn:aws:controltower:eu-west-1::control/CXUKMN5NXB00
• NIST 800-53 Rev 5 SC-7(9) | | | • Europe (London) arn:aws:controltower:eu-west-2::control/RDKAXQVBPSE
• NIST 800-53 Rev 5 SI-7(8) | | | • Europe (Stockholm) arn:aws:controltower:eu-north-1::control/DLWGUVA5BKG
• PCI DSS version 3.2.1 10.1 | | |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/QHZRCUZCBFVR</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/CXEXLMCYDMPY</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/XRXGFZGEYPMV</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/QUFDDHHUWEED</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/KBRRHKUVKJIV</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/NYHHSONNHJW</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/BFEHJBXPOGJN</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/YVQNVHXXOYLS</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/TBPNGZBEDTST</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/WHCWMAHTSOPR</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/IIIOFVZPPRKF</td>
</tr>
</tbody>
</table>
| Middle East (Bahrain)       |                    |                   | arn:aws:controltower:me
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.ATHENA.PR.1    |           | Encrypt data at rest | US East (N. Virginia) <br>arn:aws:controltower:us-east-1::control/SIBMPUMBLODX <br>US East (Ohio) <br>arn:aws:controltower:us-east-2::control/GEXGKIOWZMFF <br>US West (Oregon) <br>arn:aws:controltower:us-west-2::control/HVZCGINGZUAN <br>Canada (Central) <br>arn:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/SWNRBVQJSTPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/IPFBKEYTLWHO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/XZCPAQEMUOBV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/SOIGIAYMTJZT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/AMYQFQSBYIYG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/XWOQHXMLRFB5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/RQLNWYVGQNGR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/J0VCVNZSXVOG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/BUANQGPDSDK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/TTCZJBXUAKIS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/NEFDJXCQIHRK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/CGNIYDPMDQFA</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.ATHENA.PR.2     |           | Encrypt data at rest | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/  
LNAASCWIYQWF  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/  
BNBFJQSJJIFX  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/  
FKFIKKLBQLRU  
• Canada (Central)  
ar:n:aws:controltower:ca-central-1::control/  
IAMMTTGPSEKX  
• Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/  
UICTHPHIPSTT  
• Asia Pacific (Singapore)  
ar:n:aws:controltower:ap-southeast-1::control/  
XMSFGWXQSOWH  
• Europe (Frankfurt)  
ar:n:aws:controltower:eu-central-1::control/  
EADNJJFRAYKUJ  
• Europe (Ireland)  
ar:n:aws:controltower:eu-west-1::control/  
AZHMIHETZDO  
• Europe (London)  
ar:n:aws:controltower:eu-west-2::control/  
PUSAPFIQCZNX |

- NIST 800-53 Rev 5 CA-9(1)  
- NIST 800-53 Rev 5 CM-3(6)  
- NIST 800-53 Rev 5 SC-13  
- NIST 800-53 Rev 5 SC-28  
- NIST 800-53 Rev 5 SC-28(1)  
- NIST 800-53 Rev 5 SC-7(10)  
- NIST 800-53 Rev 5 SI-7(6)  
- PCI DSS version 3.2.1 3.4  
- PCI DSS version 3.2.1 8.2.1  

- Asia Pacific (Melbourne)  
ar:n:aws:controltower:ap-southeast-4::control/  
DSLSVZIDXAJ
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eunorth-1::control/MPGNZBGWNFVS</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/XZSYTUWYWA</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/JPLEEOSQPVEV</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/KPLQRIMGQFV</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:europe-west-3::control/XXDCQWYZWYF</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:saeast-1::control/ZETBPDEAOXPT</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/VQPTJGDPPKXK</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/FSUYYWAVPGK</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/QRXHOXQVYOA</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/SVHRKRGSYQKT</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eusouth-1::control/REMTAGETRZSC</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.AUTOSCALING.PR.1 | • NIST 800-53 Rev 5 CP-10  
• NIST 800-53 Rev 5 CP-2(2)  
• NIST 800-53 Rev 5 CP-6(2)  
• NIST 800-53 Rev 5 SC-36 | • Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
LHUKYNEKRWTH  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
IBUFUPGANKZJ  
• US West (Oregon)  
arn:aws:controltower:us-1637 |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td><a href="#">west-2::control KFRLNAHWVNDO</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Canada (Central) <a href="#">arn:aws:controltower:ca-central-1::control SDBNXVSDHXFD</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney) <a href="#">arn:aws:controltower:ap-southeast-2::control GUKAWPW0AMGJ</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore) <a href="#">arn:aws:controltower:ap-southeast-1::control QYEYJCSWQXXD</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt) <a href="#">arn:aws:controltower:eu-central-1::control LQODNIWFNOQQ</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland) <a href="#">arn:aws:controltower:eu-west-1::control MOLUIXBLTNW</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London) <a href="#">arn:aws:controltower:eu-west-2::control YJQLIPRDBJSM</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm) <a href="#">arn:aws:controltower:eu-north-1::control POOFLSRCMVHC</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai) <a href="#">arn:aws:controltower:ap-south-1::control UWTJAKCQQDQS</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul) <a href="#">arn:aws:controltower:ap-northeast-2::control FSXLENXSGMGD</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo) <a href="#">arn:aws:controltower:ap-northeast-1::control UYJOBTDPHAIM</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris) <a href="#">arn:aws:controltower:eu-west-3::control LURITYKMZLDH</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo) <a href="#">arn:aws:controltower:sa-1638</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>east-1::control/DTNVACZLNIPM</td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/NCYEGHVMTPX</td>
<td>Control API identifiers, by Region</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/EUMJCOCQOPZV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/BXWJNMFCODYNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/MOPBPZRKDT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/HDSPVQSLZUA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/LPDQXNXJOCUG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/RFZEANHEBLOT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/ISDBSSNARIHR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/YUDNGTLYTVYYU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/BPGHYVYPPDHUM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/PMSUPKWNJRRZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.AUTOSCALING.PR.2</td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td>• Protect configurations</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/GZBGNOJSJNHE</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/FIAZJHJSVVLM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/AJUNDFQNUVTWW</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:central-1::control/ZRMHGMLRJGDN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/FWWMDZRSYOB</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/WASLCDWKGJBP</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/PKWQZWEVUDPMC</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/UBAHWPQGLVLPQ</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/JZYHTBOMKDXG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/WYHXVTOLLXXU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ZSFUPIZUKOLS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/TBBXRAYTVDEK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/DUPFUTISTCKO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/RIBMFYNUIJWGF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/IJFUBJAPHSUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/INKEXBGWVWF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/MREILTLELYSG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/OKTFVJFN0EGH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/EPLQWROCHULU</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Europe (Milan) | arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.AUTOSCALING.PR.3| NIST 800-53 Rev 5 CA-9(1) | Protect configurations | • US East (N. Virginia)  
|                    | NIST 800-53 Rev 5 CM-2          |                        | arn:aws:controltower:us-east-1::control/ZSHOABWRTUIY |
|                    |                              |                        | • US East (Ohio)          
<p>|                    |                              |                        | arn:aws:controltower:us-east-1::control/ZSHOABWRTUIY |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| • NIST 800-53 Rev 5 CM-2(2)  
• PCI DSS version 3.2.1  
7.1.1                  | east-2::control/  
XDFSVFABUXJV  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/  
KQGVJSJMACGQK  
• Canada (Central)  
ar:n:aws:controltower:ca-central-1::control/  
UJLDJUJHLUYQ  
• Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/  
PDKMLRURFDNY  
• Asia Pacific (Singapore)  
ar:n:aws:controltower:ap-southeast-1::control/  
VPG0UEK3YCVR  
• Europe (Frankfurt)  
ar:n:aws:controltower:eu-central-1::control/  
ZFWZDUUOFWRV5  
• Europe (Ireland)  
ar:n:aws:controltower:eu-west-1::control/  
DZCHIICHZPIG  
• Europe (London)  
ar:n:aws:controltower:eu-west-2::control/  
MRCYHSBHWSPF  
• Europe (Stockholm)  
ar:n:aws:controltower:eu-north-1::control/  
UZQUBHHSSSTLR  
• Asia Pacific (Mumbai)  
ar:n:aws:controltower:ap-south-1::control/  
XTCCWNVXZTQQJ  
• Asia Pacific (Seoul)  
ar:n:aws:controltower:ap-northeast-2::control/  
DPCTDUCUBSOH  
• Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-northeast-1::control/  
AEBSYPXCLBY  
• Europe (Paris)  
ar:n:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>west-3::control/</td>
<td></td>
<td></td>
<td>XEASQSEGQGIK</td>
</tr>
<tr>
<td>• South America</td>
<td></td>
<td></td>
<td>(São Paulo) arn:aws:controltower:sa-east-1::control/PLLJSTBTFTPUX</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td>California) arn:aws:controltower:us-west-1::control/QQNJVLKQIYA</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Hong Kong) arn:aws:controltower:ap-east-1::control/MVBSWFPHTKLL</td>
</tr>
<tr>
<td>• Asia Pacific (Jak</td>
<td></td>
<td></td>
<td>ata) arn:aws:controltower:ap-southeast-3::control/TAVCFHICT0OC</td>
</tr>
<tr>
<td>• Asia Pacific (Osa</td>
<td></td>
<td></td>
<td>ka) arn:aws:controltower:ap-northeast-3::control/TCTWXWX5MPMI</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/ESETQPIYIQO</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/BXKGYNSCEIQO</td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td></td>
<td>(Bahrain) arn:aws:controltower:me-south-1::control/OPUWNAQIOALE</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/PHUWEEOCDRBL</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/HGYPZVXWSHAZ</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/TOHLHAQRVXXW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.AUTOSCALING.PR.4 | • NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 CP-2(2)  
• NIST 800-53 Rev 5 SI-2  
• PCI DSS version 3.2.1  
• PCI DSS version 2.2 | • Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/UKYGXEVHJUTL  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/WMWBOCEROGJK  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/UEEYNVTGUBLF  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/VCDSRAXDZLZP  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/DIXWASPFN0FQ  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/WAVCCMIDCEXH  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/POLAIJLNEGQU |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/IYKSVEZZLTNF</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/TJGWWGICHSES</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/NVNXTLBWMUW</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/SOTBXAWRFZB</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/ZFEQJGAMVZT</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/WWETVQMUSGFB</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/HHTBCTOYPGHZ</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/MIHQSUSOAQN</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/0DABNVXWCCWN</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/XJPQIVKNHPFP</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/XVBVPLVRQXBC</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Osaka) |          |                   | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.AUTOSCALING.PR.5</td>
<td>• NIST 800-53 Rev 5 AC-21</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-1647</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 AC-3</td>
<td>east-1::control/GXOHAMKZBEH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/RHEYJDCZKXSD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 AC-4</td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/IKGGRITHSTQX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 AC-4(21)</td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/CXHVX0ATJHPY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 AC-6</td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AQIXLPCRCRQHS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7</td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/THTGVEGTGXB0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/GDYUFOPBQXQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/KICFAVQMR0F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7(20)</td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/UPTNYPJYCVQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7(21)</td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/CZLFOZMNVYS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7(3)</td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/YWRLTVCFFEXT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/MOWUHAQ08RJY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7(9)</td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>northeas-1::control/RJDVSNZ0OQNV</td>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/IHWPSXMEXIFN</td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/IHWPSXMEXIFN</td>
<td>arn:aws:controltower:eu-west-3::control/1HWPSXMEXIFN</td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/NPDLLNLLZAFAC</td>
<td>arn:aws:controltower:sa-east-1::control/NPDLLNLLZAFAC</td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/NPDLLNLLZAFAC</td>
<td>arn:aws:controltower:sa-east-1::control/NPDLLNLLZAFAC</td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/PKLTMQWOKPC</td>
<td>arn:aws:controltower:us-west-1::control/PKLTMQWOKPC</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ESHYEDTXWUSW</td>
<td>arn:aws:controltower:ap-east-1::control/ESHYEDTXWUSW</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/MFBWYIYZRZWU</td>
<td>arn:aws:controltower:ap-southeast-3::control/MFBWYIYZRZWU</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/FOSLERMMWCCQ</td>
<td>arn:aws:controltower:ap-northeast-3::control/FOSLERMMWCCQ</td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/LZHHRTKSKBYG</td>
<td>arn:aws:controltower:eu-south-1::control/LZHHRTKSKBYG</td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/YIJBLHBBQ2ZV</td>
<td>arn:aws:controltower:af-south-1::control/YIJBLHBBQ2ZV</td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/HWYCNKQJIZT</td>
<td>arn:aws:controltower:me-south-1::control/HWYCNKQJIZT</td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/DZZROKPAAXCY</td>
<td>arn:aws:controltower:il-central-1::control/DZZROKPAAXCY</td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/YJUEUNVOJKA0</td>
<td>arn:aws:controltower:eu-central-2::control/YJUEUNVOJKA0</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.AUTOSCALING.PR.6 | • NIST 800-53 Rev 5 CP-10  
• NIST 800-53 Rev 5 CP-2(2)  
• NIST 800-53 Rev 5 CP-6(2)  
• NIST 800-53 Rev 5 SC-36  
• NIST 800-53 Rev 5 SC-5(2)  
• NIST 800-53 Rev 5 SI-13(5) | • Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/IMJNNSRJLUDS  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/SBFDHQUTINU  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/XMQUQEMIYMAQ  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/KNWXLCUZBJPR  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/HYEMAFKYGEOG  
• Asia Pacific (Singapore)  
arwn:aws:controltower:ap-southeast-1::control/JBBGLEZGYNNY |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/PFKPZDJXQARN</td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/WMTUSTWLNDZO</td>
</tr>
<tr>
<td>Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/ZPCRFNLDIVTB</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/EQRHGKGNQNGX</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/OLNBKFLQLQMBM</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/PALLYLIKPD0O</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/RIHZBJGCQNZN</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/VVHBDVAENCMX</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/RGGVMSFSSHVVI</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/VULHVTLCFHLY</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/EVLGNFFONEDR</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>southeast-3::control/RIAGHUSPAQBY</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/BSHZJGOWIYAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/WTWCPKFEWVYK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/NCNVQPRQZRO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/DWSDKOZNKIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/QAQRWYEQZGK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/UGALPVXQVQC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/LEMBCAXJDJXF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/NCQS0ESJIZPDJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/HLLQBYJSJH0V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/EWEADHPWKAQO</td>
</tr>
</tbody>
</table>
## CT.AUTOSCALING.PR.8

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.AUTOSCALING.PR.8 | • NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-2  
• NIST 800-53 Rev 5 CM-2(2)  
• PCI DSS version 3.2.1  
2.2 | • Manage vulnerabilities | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/JAZMSKNXGBDD  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/CBEYLWICUQCE  
• US West (Oregon)  
arn:aws:aws:controltower:us-west-2::control/PKAHSCIYKNJT  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/SUQITVFXSNAS  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/ZBBMSBCWHDJT  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/CTWWGOPBJZZV  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/USQMZZGHDRYN  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/YBNINIYMRBIYN  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/ZYAHAGQKQTAN  
• Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/JHZVIDCOFKWQ  
• Asia Pacific (Mumbai)  
arn:aws:controltower:ap-south-1::control/JU1BGGNJBJHVD |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>Arn:aws:controltower:ap-northeast-2::control/YZYBKZLRYPLJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>Arn:aws:controltower:ap-northeast-1::control/LIFOKTEDIUYJG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>Arn:aws:controltower:eu-west-3::control/DQVZDZGYRISN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>Arn:aws:controltower:sa-east-1::control/GVXMKGJZQXQZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>Arn:aws:controltower:us-west-1::control/PISDHZLSRYXL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>Arn:aws:controltower:ap-east-1::control/LMMGXHIXUJYZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>Arn:aws:controltower:ap-southeast-3::control/HFDQZKOIUTPA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>Arn:aws:controltower:ap-northeast-3::control/UBFEQJTFABY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>Arn:aws:controltower:eu-south-1::control/YTPNKLYIKVY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>Arn:aws:controltower:af-south-1::control/YVPFFBYVWOYU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>Arn:aws:controltower:me-south-1::control/AEHEFPZQLWU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>Arn:aws:controltower:il-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.CLOUDFORMATION.PR.1 | None | • Protect configurations | central-1::control/SACRRCECVZND  
• Europe (Zurich)  
ar:aws:controltower:eu-central-2::control/QIRLBHLADKAB  
• Europe (Spain)  
ar:aws:controltower:eu-south-2::control/PSRUDUBINGKQS  
• Asia Pacific (Hyderabad)  
ar:aws:controltower:ap-south-2::control/DKHSPQGATGW  
• Middle East (UAE)  
ar:aws:controltower:me-central-1::control/FJYUYTCKFRPF  
• Asia Pacific (Melbourne)  
ar:aws:controltower:ap-southeast-4::control/CVLWHPNFDHV |

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.CLOUDFORMATION.PR.1 | None | • Protect configurations | US East (N. Virginia)  
ar:aws:controltower:us-east-1::control/WTDSMKDKDNLE  
US East (Ohio)  
ar:aws:controltower:us-east-2::control/FMBVJTKQAEAE  
US West (Oregon)  
ar:aws:controltower:us-west-2::control/DKKMAWVORGDV  
Canada (Central)  
ar:aws:controltower:ca-central-1::control/OMCTIJOASMIZ  
Asia Pacific (Sydney)  
ar:aws:controltower:ap- |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>southeast-2::control/ UHAFTVUQLBJQ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-southeast-1::control/ VQJENYWQLIKN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-central-1::control/ IEBSHSUWVLNW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-west-1::control/ XLSIRLRDKWVQ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-west-2::control/ CPQMWUWNTTQN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-north-1::control/ RTHCEFFNCBIC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-south-1::control/ ZNYKRMJOMGMD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-northeast-2::control/ AZAYRJYXCRZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-northeast-1::control/ TUJJPJITMNX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-west-3::control/ PXUHPQAIVTCU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:sa-east-1::control/ QZWDNRQVJIKO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/ KDYIRULEPEVX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-east-1::control/TRDTWWMRSUZAV</code></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-southeast-3::control/FZACIPKYOZWJ</code></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-northeast-3::control/BKGHDLNFOLGZ</code></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:eu-south-1::control/QVSBUPSGUHVS</code></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:af-south-1::control/OBRUJGFHLWTL</code></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:me-south-1::control/RBUJYPEKBWAQ</code></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:il-central-1::control/XWTPZYNKDZAN</code></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:eu-central-2::control/ZYKOUZYVAXVX</code></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:eu-south-2::control/SUZUXRSEEELFT</code></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-south-2::control/RTJVHAXOEEDEY</code></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:me-central-1::control/YKZGVTBPBQF</code></td>
</tr>
<tr>
<td>Asia Pacific (Melbourne)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-</code></td>
</tr>
</tbody>
</table>
## Control identifier | Framework | Control objective | Control API identifiers, by Region
---|---|---|---
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 AC-17(2) | Protect configurations | US East (N. Virginia) arn:aws:controltower:us-east-1::control/XDNLXQXCMXXF
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 AC-4 | | US East (Ohio) arn:aws:controltower:us-east-2::control/IQSRCUCSGFPK
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 IA-5(1) | | US West (Oregon) arn:aws:controltower:us-west-2::control/AVEFYTVGFGCS
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 SC-12(3) | | Canada (Central) arn:aws:controltower:ca-central-1::control/TQHOWLDDEHMD
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 SC-13 | | Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/HGEUZPXSVDO5
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 SC-23 | | Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/PHMWMCANHKSW
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 SC-7(11) | | Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ZWIHYGTQCIQA
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 SC-7(16) | | Europe (Ireland) arn:aws:controltower:eu-west-1::control/UAUKDTHXFEXN
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 SC-7(4) | | Europe (London) arn:aws:controltower:eu-west-2::control/KRZLJNJOIPEAE
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 SC-8 | | Europe (Stockholm) arn:aws:controltower:eu-north-1::control/MYOOHMLVKUVG
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 SC-8(1) | | |
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 SC-8(2) | | |
CT.CLOUDFRONT.PR.1 | NIST 800-53 Rev 5 SI-7(6) | | |
CT.CLOUDFRONT.PR.1 | PCI DSS version 3.2.1 1.2.1 | | |
CT.CLOUDFRONT.PR.1 | PCI DSS version 3.2.1 1.3 | | |
CT.CLOUDFRONT.PR.1 | PCI DSS version 3.2.1 1.3.1 | | |
CT.CLOUDFRONT.PR.1 | PCI DSS version 3.2.1 1.3.2 | | |
CT.CLOUDFRONT.PR.1 | PCI DSS version 3.2.1 1.3.4 | | |
CT.CLOUDFRONT.PR.1 | PCI DSS version 3.2.1 1.3.7 | | |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controllower:ap-south-1::control/JURDJLNESWR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controllower:ap-northeast-2::control/VBJDOHQSIAX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controllower:ap-northeast-1::control/XPNZKJLPXQA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controllower:eu-west-3::control/ITCBSQBOBUDP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controllower:saeast-1::control/XOWFTDERUYHD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controllower:us-west-1::control/IWQIWQREBLHB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controllower:ap-east-1::control/MKJZCQJTJUU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controllower:ap-southeast-3::control/TOOGKXNIZTGI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controllower:ap-northeast-3::control/KFBJHZDCITRT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controllower:eu-south-1::control/MLF0FPIEQPDK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controllower:af-south-1::control/ZKNGGEQJFJEW</td>
</tr>
</tbody>
</table>
| | | | • Middle East (Bahrain) arn:aws:controllower:me-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.CLOUDFRONT.PR.10 | ・ NIST 800-53 Rev 5 SC-7(11)  
・ PCI DSS version 3.2.1 1.2.1  
・ PCI DSS version 3.2.1 1.3  
・ PCI DSS version 3.2.1 1.3.1  
・ PCI DSS version 3.2.1 1.3.2  
・ PCI DSS version 3.2.1 1.3.4 | ・ Enforce least privilege  
・ Protect configurations | ・ US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/  
MKKANNMZZHNU  
・ US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/  
WCTTAHVMYDLS  
・ US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/  
IJBYZPYEBBPL  
・ Canada (Central)  
ar:n:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>central-1::control/EMBBJBOEZUXI</td>
<td>• Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/XSFJGHXKXXJQ</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/TJJCFQZERMMU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/DPTTYLBMYLES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/COJLJAOYIIY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/EDDBQAMFSHML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/KEBVYBUGDCJL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/DPDNFQCKPXR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/YBHTLHKEKZF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/RFCFGQXJZARA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/JSCPKBPTUFX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:saeast-1::control/XHRXFSGXWPKZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/GQNIWWFCXLMC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/RMFPWFTBHEUW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/RNEBJBKTWYNP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/HPIJWRRRNZWV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/ZSTTVOTUCTJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/QOUFOYZMAHTP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/CRMMLULUZQFR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/KKJWKODDJAZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/RIZVPMYJKMF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/UTNKPVFYDNYWY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/NCQFJEHLNVOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/NREUZZCXLORX</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.CLOUDFRONT.PR.11</td>
<td>NIST 800-53 Rev 5 AC-17(2)</td>
<td>Manage vulnerabilities</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/JKSWKPYLWXBN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/FRBHAALKXVJQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 IA-5(1)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/YXRLGXTKHYT</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-12(3)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ROXOJWPKIJG</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/OLKKBIYGTTT</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-23</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/PWKKNYCLUDT</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/00IGLWNRLDBR</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/IAUDCMRWXWEY</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:europe-north-1::control/OEQBOVEXMMKR</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:asia-pacific-south-1::control/LATRCFDQWVIN</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:asia-pacific-northeast-2::control/ICESDVPQUNI</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:asia-pacific-northeast-1::control/HFHBWIGNGHMY</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:europe-west-3::control/HAPUSNWOPEXI</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:south-america-east-1::control/RAYNZKHYBPAB</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:us-west-1::control/VDNOLOGKWONY</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:asia-pacific-east-1::control/RKEOFHPSMRG</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:asia-pacific-southeast-3::control/EBMFFONQCPFD</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:asia-pacific-northeast-3::control/VZFSUGKCIKSL</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:europe-south-1::control/ACOALEKSQBDV</td>
</tr>
</tbody>
</table>
| Africa (Cape Town) |       |                  | Arn:aws:controltower:afr-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.CLOUDFRONT.PR.2 | NIST 800-53 Rev 5 SC-7(11) | Protect configurations | · US East (N. Virginia) arn:aws:controltower:us-east-1::control/WPBVZXCJXPXBZ  
· US East (Ohio) arn:aws:controltower:us-east-2::control/MLHMBKFPSZQT  
· US West (Oregon) arn:aws:controltower:us-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>west-2::control/UTWBDZNQLUD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Canada (Central)arn:aws:controltower:ca-central-1::control/RUMPJMCCNCJW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)arn:aws:controltower:ap-southeast-2::control/NMGEPJROACOH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)arn:aws:controltower:ap-southeast-1::control/JPIDQBIFWBBW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt)arn:aws:controltower:eu-central-1::control/YIXCHURUJDBK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland)arn:aws:controltower:eu-west-1::control/SPJYDXWVBVHQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London)arn:aws:controltower:eu-west-2::control/TZHFMCVNUQB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm)arn:aws:controltower:eu-north-1::control/AKXTQWFSQTVQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)arn:aws:controltower:ap-south-1::control/SDPQNYINEQRA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)arn:aws:controltower:ap-northeast-2::control/I00JYJOANCXR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo)arn:aws:controltower:ap-northeast-1::control/BMFRHKRDLPYE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris)arn:aws:controltower:eu-west-3::control/VTSTOPJEHAKZN</td>
</tr>
</tbody>
</table>
| | | | South America (São Paulo)arn:aws:controltower:sa-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>east-1::control/SGPGHMEKOYTP</td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/SBPRUJEJCGTL</td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/JYTOHOYSNLMB</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-southeast-3::control/SLGAANKAFEYA</td>
<td>• Europe (Milan) arn:aws:controltower:eusouth-1::control/TOJRWDYHNNXG</td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/JTUVGABKLAFE</td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain) arn:aws:controltower:messouth-1::control/SEODAVXUYYYKF</td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/ENWFNBADPKGV</td>
<td>• Europe (Zurich) arn:aws:controltower:eucentral-2::control/XKAFLKSMRCTZ</td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain) arn:aws:controltower:eusouth-2::control/HPURLNWRKJWL</td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:apsouth-2::control/KPXXRYKIBWCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>CT.CLOUDFRONT.PR.3</td>
<td>NIST 800-53 Rev 5 AC-17(2)</td>
<td>Encrypt data in transit</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/KSCPLCGJ3KGHW</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/RFNWXVMMUBPQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 IA-5(1)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/LHFZBEUHUBMH</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-12(3)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/YXRNFYZICBAZ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/MCQETJVLFUOU</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-23</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/JLXONOZWHECI</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-23(3)</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/OIWLNKZULIHF</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/AGJSNVMSVMZW</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/GZSHUDGCTKTD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/KFVEGDRDKNYI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/OCRRDMBCTTP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/MKLFH1IEWWR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/TMYDUSKTAKBS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/YDQTLQWZBPXO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/LJBMFILEKDJH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/QSWVHIUPCHAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/DDVLBLNWHGDUB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/ISFZIBFIGEYE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/YQYFXRCGOJVL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.CLOUDFRONT.PR.4 | • NIST 800-53 Rev 5 CP-10  
• NIST 800-53 Rev 5 SC-36 | • Improve availability | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/  
TTGXTDSTFUVR |
|                    |          |                  |        | • US East (Ohio)  
ar:n:aws:controltower:us-east-1::control/  
TTGXTDSTFUVR |
|                    |          |                  |        | • Middle East (Bahrain)  
ar:n:aws:controltower:me-south-1::control/  
SQZNZFSBBJQD |
|                    |          |                  |        | • Asia Pacific (Hyderabad)  
ar:n:aws:controltower:ap-south-2::control/  
NJQTYERBVBJI |
|                    |          |                  |        | • Middle East (UAE)  
ar:n:aws:controltower:me-central-1::control/  
HICXKNXMCXZ |
|                    |          |                  |        | • Asia Pacific (Melbourne)  
ar:n:aws:controltower:ap-southeast-4::control/  
TYWUXNEBHG |

**Control identifier**

**Framework**

**Control objective**

**Control API identifiers, by Region**

- **Africa (Cape Town)**
  arn:aws:controltower:af-south-1::control/ EOVQXRLMTRYED
- **Middle East (Bahrain)**
  arn:aws:controltower:me-south-1::control/ SQZNZFSBBJQD
- **Israel (Tel Aviv)**
  arn:aws:controltower:il-central-1::control/ AFQUJTLNZOZD
- **Europe (Zurich)**
  arn:aws:controltower:eu-central-2::control/ LRZZVKQUSKFK
- **Europe (Spain)**
  arn:aws:controltower:eu-south-2::control/ CVMBTUQQJQVF
- **Asia Pacific (Hyderabad)**
  arn:aws:controltower:ap-south-2::control/ NJQTYERBVBJI
- **Middle East (UAE)**
  arn:aws:controltower:me-central-1::control/ HICXKNXMCXZ
- **Asia Pacific (Melbourne)**
  arn:aws:controltower:ap-southeast-4::control/ TYWUXNEBHG
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td>east-2::control/BCDVICOTHADR</td>
<td></td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/RFYXXYNURHWK</td>
<td></td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/CXLHFJBBESPG</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/ELYJMXPJVKEI</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/RADLRUNIIIXJV</td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/PPDPQYKYCVQV</td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/WJKMTILNPQPE</td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/LEHRPQBGFKSI</td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/KLXWEEBCGMDJ</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/HPOIOKYDFMZ</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/GFFYUYUVPDKU</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/VOOZHGULFVBG</td>
<td></td>
</tr>
</tbody>
</table>
| • Europe (Paris) | | arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>west-3::control/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GWBOIBMAAVZC</td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LULEEVQWEQRC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IUIGCLTOJBUC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WJKRCBQFHZMZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>southeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IPCYTHQFAFNW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>northeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DOTFJQVJFLRM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OEGLEOAAPISR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:af-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VXRYQQGVHSUW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:me-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BSHPWYCFNNBE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:il-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LLUWIDQCLSLL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>central-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OMEWBMVIXJXG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>south-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LEXJMNWZMLIVI</td>
</tr>
</tbody>
</table>
### CT.CLOUDFRONT.PR.5

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.CLOUDFRONT.PR.5</td>
<td>• NIST 800-53 Rev 5 AC-2(4)</td>
<td>• Establish logging and monitoring</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/RSBKBJHWOCY</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(26)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/EMCHZAZDZNVL</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6(9)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/MAPJFIKMKJ3KD</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ZYDBRRMVVTZV</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/COFQNFYPVILDJ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/RHBUCASOCZAV</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/XZRMWQOAQCCXL</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/ RSSXGIZAPHTX</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ NWKZMIUZNFKT</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ JNBLGPSROEFQ</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ QDMUYVEFHYOR</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ CGZWLZPDWIMX</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ EDSGIRTZHTA</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ CZAGHRZSABRN</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ RXASZFBDCWEX</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ TBJTNHJFQFRV</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ HGFGDWIJHIUM</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ XRYUQXKTPXYZ</td>
<td></td>
</tr>
</tbody>
</table>
| • PCI DSS version 3.2.1 | Asia Pacific (Osaka) | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.CLOUDFRONT.PR.6 | • NIST 800-53 Rev 5 AC-17(2) | • Encrypt data in transit | • US East (N. Virginia) arn:aws:controltower:us-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
<td>east-1::control/QFQNQHNDSHBP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 IA-5(1)</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/LQDI0WJWUTSM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-12(3)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/EOUIZKHSXPN</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/LLIUTEJBMGHE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-23</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/IYBPXPFECFI</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-23(3)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/TUIUKMPMAEUD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/MAELDXXMVECM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/OBRSEKOWVRVB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(1)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/NJKKOAWYJJM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/LOAXFPPQRMCF</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/SWGGZGILWLV5</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/MCPECQPPQYPPZ</td>
</tr>
</tbody>
</table>
| • PCI DSS version 3.2.1 4.1 | | | Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeast-1::control/ YFRQGLAWSBZU</td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/ APQPRHGKFWYO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/ WMHKGFLJAWPI</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/ HYJHVPFFKHDY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/ LEONXWJLQTYM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/ SAEMWRDHDRO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/ WBMVRQPARQX</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/ HYECM3HMDQAO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/ MEVRARVQYQCB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/ QWHDIDDVRIX</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/ RLNLTYXPMQNX</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/ LTKLSWPOPLNR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.CLOUDFRONT.PR.7</td>
<td></td>
<td></td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/ SFIAOWRGHFHB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/ ZQXCMIVMKIAA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/ SUXMGZJXERLV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/ EYUKXNOQHBVC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/ AQDUDHILVQNE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/ XFODQONNCMNG</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/FLFNIUITMQCJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/PJOCVKGBKJLJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/HVCNXRUCMQUQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/AMBUPGBVMSRB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/BYQGXDIVQDFM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/KYIFISGDQAHH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/PVAWACSEUZOC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/VKCCNJKDKWNI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/TKNLPUTAFIBQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/XDXVMJWXNHM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/FOMSEIIEPEIR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-1679</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>southeast-3::control/FEOXIQBHYQON</td>
<td>• Asia Pacific (Osaka)</td>
<td><em>arn:aws:controltower:ap-northeast-3::control/UHYZMCZNNITK</em></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td><em>arn:aws:controltower:eu-south-1::control/LANXJNMROUBG</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td><em>arn:aws:controltower:af-south-1::control/OIQLPXBBDAEF</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td><em>arn:aws:controltower:me-south-1::control/JDWMKBYCFRRZ</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td><em>arn:aws:controltower:il-central-1::control/VQBHQ0YFACWB</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td><em>arn:aws:controltower:eu-central-2::control/UETNVSJMZFBSB</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td><em>arn:aws:controltower:eu-south-2::control/CUEQJGENOTAN</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td><em>arn:aws:controltower:ap-south-2::control/UPTABJLZLULD</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td><em>arn:aws:controltower:me-central-1::control/RANNJLPZMAWZ</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Melbourne)</td>
<td><em>arn:aws:controltower:ap-southeast-4::control/XZFLXXIQYNUD</em></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CT.CLOUDFRONT.PR.8

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.CLOUDFRONT.PR.8 | • NIST 800-53 Rev 5 AC-17(2)  
• NIST 800-53 Rev 5 AC-4  
• NIST 800-53 Rev 5 IA-5(1)  
• NIST 800-53 Rev 5 SC-12(3)  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-23  
• NIST 800-53 Rev 5 SC-23(3)  
• NIST 800-53 Rev 5 SC-7(4)  
• NIST 800-53 Rev 5 SC-8  
• NIST 800-53 Rev 5 SC-8(1)  
• NIST 800-53 Rev 5 SC-8(2)  
• NIST 800-53 Rev 5 SI-7(6)  
• PCI DSS version 3.2.1 4.1 | • Encrypt data in transit | • US East (N. Virginia)  
ar:aws:controltower:us-east-1::control/YPZYCTALWZLK  
• US East (Ohio)  
ar:aws:controltower:us-east-2::control/KQTKKFCMSTJR  
• US West (Oregon)  
ar:aws:aws:controltower:us-west-2::control/DIFDZPKTUXZR  
• Canada (Central)  
ar:aws:controltower:ca-central-1::control/FZHDAFBJMXP  
• Asia Pacific (Sydney)  
ar:aws:controltower:ap-southeast-2::control/FWXCSWFYXMC1  
• Asia Pacific (Singapore)  
ar:aws:controltower:ap-southeast-1::control/XKYWAPJWATKR  
• Europe (Frankfurt)  
ar:aws:controltower:eu-central-1::control/VCVFZRI3CQ7Z  
• Europe (Ireland)  
ar:aws:controltower:eu-west-1::control/CNNZUYODKIDH  
• Europe (London)  
ar:aws:controltower:eu-west-2::control/YPHH5QJLYW0Q  
• Europe (Stockholm)  
ar:aws:controltower:eu-north-1::control/HLXS0ZZDWMJN  
• Asia Pacific (Mumbai)  
ar:aws:controltower:ap-south-1::control/CFFXQUBT6GPR |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/DNSOBLSOWZBO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/VSBTG00GYBBH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ONPPIEAALZUC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/APWMSFWSKMFJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/WIVHPNTBEMAA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/RTJQECKEXLQP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/UCOWSXPWKCCI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/FFUABXMEWYYC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/JLXINGWIDTUJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/ROHXSITDLDWI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/XBLMALRLBRNO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Israel (Tel Aviv) | arn:aws:controltower:il-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.CLOUDFRONT.PR.9</td>
<td>NIST 800-53 Rev 5 AC-17(2)</td>
<td>- Manage vulnerabilities</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/XCQMS00MO0KF</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/HXXCDBIITXOE</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-12(3)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:central-1::control/0JPV1MZHKWMN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-23</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-23(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(1)</td>
<td></td>
<td></td>
<td>southeeast-2::control/MRNNWKBUPWpW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/ORPMLBIMJYEY</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/DVICCSXHKOE</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/ADCFMZEXRPH</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/BJDRLQZJKZL</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/IZKULFXRTZNU</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/NBUGVULZANXD</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/PZBIGNJZQZWK</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/GITAQKSEJDXV</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/EUOWQLXWFGNQ</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:saeast-1::control/AHXYZHBAZBFH</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/HXNZJSDBYUQE</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/KYTE0GXXU3PF</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/VFZMJZENUMGT</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/LFPEMXYIZG0G</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/HYDVIHSHNUJK</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/UMSU00DGJNE</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/MSBIHIZOYSWP</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/LCDMMMAPSTOM</td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/YMJHSAINUZQL</td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/MVBZHLFJEXXG</td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/HJIQHQJ2ZMQZ</td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/YTPDUMS1BEB0</td>
</tr>
</tbody>
</table>
| Asia Pacific (Melbourne) | | | arn:aws:controltower:ap-
### CT.CLOUDTRAIL.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.CLOUDTRAIL.PR.1</td>
<td>• CIS AWS Benchmark 1.4.3.7</td>
<td>• Encrypt data at rest</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-9</td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/EPNCREENAXLY</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/CTXDBBBXWFYM</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/RBOKYZWNIEEH</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/LADQHOPWFDQE</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/MQTVMQFRAQQV</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/LDWQNCXTIACQ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/QHBOXOKOYNIQV</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/GNPFAAZFPLUN</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.5</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ZWQEGGDWHBMQ</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.5.2</td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/FRHFJVRAPTRV</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
</tbody>
</table>
| Asia Pacific       |           |                   | • Asia Pacific (Mumbai)  
arwn:aws:controltower:ap-south-1::control/YVIVTCSBKNZW |
| South America      |           |                   | • South America (São Paulo)  
arwn:aws:controltower:sa-east-1::control/AZGUTSGKCXJA |
| Europe             |           |                   | • Europe (Paris)  
arwn:aws:controltower:eu-west-3::control/KDFZJGDKJDXF |
| US West            |           |                   | • US West (N. California)  
arwn:aws:controltower:us-west-1::control/CJBDBOPJMYK |
| Asia Pacific       |           |                   | • Asia Pacific (Hong Kong)  
arwn:aws:controltower:ap-east-1::control/JLCBXYLJKYT |
| Asia Pacific       |           |                   | • Asia Pacific (Jakarta)  
arwn:aws:controltower:ap-southeast-3::control/JQRQQRXWACD |
| Asia Pacific       |           |                   | • Asia Pacific (Osaka)  
arwn:aws:controltower:ap-northeast-3::control/XXBNGWACEEJO |
| Europe             |           |                   | • Europe (Milan)  
arwn:aws:controltower:eu-south-1::control/JHSHFWNLDOKX |
| Africa             |           |                   | • Africa (Cape Town)  
arwn:aws:controltower:af-south-1::control/CAZMRVABVNVZ |
| Middle East        |           |                   | • Middle East (Bahrain)  
arwn:aws:controltower:me- |
## CT.CLOUDTRAIL.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.CLOUDTRAIL.PR.2 |            | Protect data integrity | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/  
ZOBWKZJTOUFY  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/  
XFIDQGGGOMTT  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/  
SPIMIJLBSCTG  
• Canada (Central)  
ar:n:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 11.5</td>
<td>Central-1::control/KAEEWMTLQTBG</td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/DTÍKKBFWTRD</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/VLMNMXWTFIR</td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/RJSEMFBKORT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ABAVTAQIBMN</td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ZMDJHYXAEUWW</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/YAJIWBETKSKD</td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/XOLQDYEYNFBA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/MXWILHEXRPJ</td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/PNBCQPCGUYV</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/QCMJJEOTGECG</td>
<td>South America (São Paulo) arn:aws:controltower:saeast-1::control/GDFPDQOFIRPG</td>
<td></td>
</tr>
<tr>
<td></td>
<td>US West (N. California) arn:aws:controltower:cenral-1::control/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE)</td>
</tr>
</tbody>
</table>

### CT.CLOUDTRAIL.PR.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.CLOUDTRAIL.PR.3 | • CIS AWS Benchmark 1.4 3.4
• NIST 800-53 Rev 5 AC-2(4)
• NIST 800-53 Rev 5 AC-4(26)
• NIST 800-53 Rev 5 AC-6(9)
• NIST 800-53 Rev 5 AU-10
• NIST 800-53 Rev 5 AU-12
• NIST 800-53 Rev 5 AU-2
• NIST 800-53 Rev 5 AU-3
• NIST 800-53 Rev 5 AU-6(1)
• NIST 800-53 Rev 5 AU-6(3)
• NIST 800-53 Rev 5 AU-6(4)
• NIST 800-53 Rev 5 AU-6(5)
• NIST 800-53 Rev 5 AU-7(1)
• NIST 800-53 Rev 5 AU-9(7)
• NIST 800-53 Rev 5 CA-7
• NIST 800-53 Rev 5 SC-7(9)
• NIST 800-53 Rev 5 SI-20
• NIST 800-53 Rev 5 SI-3(8) | • Establish logging and monitoring | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/DZJUTMXLWLMF
• US East (Ohio) arn:aws:controltower:us-east-2::control/NKNKEYMHFHH
• US West (Oregon) arn:aws:controltower:us-west-2::control/XAFE0VTQSTJA
• Canada (Central) arn:aws:controltower:ca-central-1::control/GKZHBZWQTXQ
• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AKRKBXPLHCR0
• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ENOFRZNYKBC
• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/DGZZTHPBJAJX
• Europe (Ireland) arn:aws:controltower:eu-west-1::control/BEFMOHMGMYG
• Europe (London) arn:aws:controltower:eu-west-2::control/ENTYLMYRYEDC |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/MFOJTEUROQZPI</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(5)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/AFDQMHYVCZPG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td>England</td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/GUMXCBJWOVUY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.3</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/BBEODQCLACOK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.4</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/HXNJHOVBCEQW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/TZTITSLDPMHU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/BCQJMEYFTIGX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/RIRMCQIOIAWT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/NDEJIGGFJKIR</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/HDBBNLSSWFJN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.CLOUDTRAIL.PR.4</td>
<td></td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/AUDOYQMVFSG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/00PJJACQ0FUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-</td>
</tr>
</tbody>
</table>

### Control identifier

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.CLOUDTRAIL.PR.4</td>
<td></td>
<td>Encrypt data at rest</td>
</tr>
</tbody>
</table>

### Framework

- NIST 800-53 Rev 5 CA-9(1)
- NIST 800-53 Rev 5 CM-3(6)
- NIST 800-53 Rev 5 SC-13
- NIST 800-53 Rev 5 SC-28

### Control objective

- Encrypt data at rest

### Control API identifiers, by Region

- US East (N. Virginia) arn:aws:controltower:us-east-1::control/AUDOYQMVFSG
- US East (Ohio) arn:aws:controltower:us-east-2::control/00PJJACQ0FUS
- US West (Oregon) arn:aws:controltower:us-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td></td>
<td>west-2::control/MIAFWDCMLTQFG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/QDLDUFIKRTAF</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/CTOCGYRSDLYB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/NCGCPHKAILMF</td>
</tr>
<tr>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/WGWSUAMMMWWN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/BIDXZLOVTAPH</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/YRILAMGJGTF5</td>
</tr>
<tr>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/GUOVVFIECKHB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/UQXTLQQUVMR0</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/RFBRGBFCHGRH</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/DJVAKGZ0BGG</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Europe (Paris) arn:aws:controltower:eu-west-3::control/HDTLDGKSVHCS | | | South America (São Paulo) arn:aws:controltower:sa-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>east-1::control/NOFOPVIFERIB</td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/ARZVFSOFFRFRN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/CFOQOBAUVZYH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/GBLIBFBFWEWJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/PRBKENAXOJXR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/CQIIWIZRIDXD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/HLHWSRJCEYDM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/UQUAZOG3VRPA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/KVTMFMCHBSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/YRWHIEUGKWK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/EQJLBPKNWCEW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/AWBLELLPBPBZ</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------</td>
<td>--------------------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>CT.CLOUDWATCH.PR.1</td>
<td>• NIST 800-53 Rev 5 AU-6(1)</td>
<td>• Establish logging and monitoring</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/SSXFCVHAAAYDY</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-6(5)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/RAMYQIPCOQPV</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/TKCJNPXIDFHI</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 IR-4(1)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/VQRWIUPQXDNG</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 IR-4(5)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/MOWBJMX0IHSB</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/EFYIPKXXIMWCM</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-20</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/0BZIWNWJNIFK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-4(12)</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/YPSCUERHMDGL</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-4(5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/YDPEPIRVMRSF</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/FDRWVVLULZAD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Control identifier**

**Framework**

**Control objective**

**Control API identifiers, by Region**
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/EDZSIQLZSHQT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/SFQHEJ3LEUAE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/VZQEAYLRUVEW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/JWZXOWQPDALC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/DZZBWHYOVWZH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/GVINKYTHSTYG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/YKLBLSNIPAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/YSCEYMW3PDFX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/JASXXRCJAGRM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/EHGRDSWJBZRD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/XFTUWJDDHWR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Europe (Milan) | arn:aws:controltower:eu-
### CT.CLOUDWATCH.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.CLOUDWATCH.PR.2</td>
<td>• NIST 800-53 Rev 5 AU-10</td>
<td>• Establish logging and monitoring</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-11</td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/HTUKCDNXPVWk</td>
</tr>
<tr>
<td></td>
<td><strong>Control identifier</strong></td>
<td></td>
<td><strong>US East (Ohio)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/HTUKCDNXPVWk</td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td><strong>Europe (Zurich)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/IGPKWKCRTQSA</td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td><strong>Europe (Spain)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/IYZCFYKTLOEZ</td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/XSCOQUSBVAKH</td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td><strong>Middle East (UAE)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/CNMHXHONGEWR</td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td><strong>Asia Pacific (Melbourne)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/FCCVJPUNPVXQ</td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td><strong>Europe (Zurich)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/IGPKWKCRTQSA</td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td><strong>Europe (Spain)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/IYZCFYKTLOEZ</td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/XSCOQUSBVAKH</td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td><strong>Middle East (UAE)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/CNMHXHONGEWR</td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td><strong>Asia Pacific (Melbourne)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/FCCVJPUNPVXQ</td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td><strong>Europe (Zurich)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/IGPKWKCRTQSA</td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td><strong>Europe (Spain)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/IYZCFYKTLOEZ</td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/XSCOQUSBVAKH</td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td><strong>Middle East (UAE)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/CNMHXHONGEWR</td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td><strong>Asia Pacific (Melbourne)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/FCCVJPUNPVXQ</td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td><strong>Europe (Zurich)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/IGPKWKCRTQSA</td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td><strong>Europe (Spain)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/IYZCFYKTLOEZ</td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/XSCOQUSBVAKH</td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td><strong>Middle East (UAE)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/CNMHXHONGEWR</td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td><strong>Asia Pacific (Melbourne)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/FCCVJPUNPVXQ</td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td><strong>Europe (Zurich)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/IGPKWKCRTQSA</td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td><strong>Europe (Spain)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/IYZCFYKTLOEZ</td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/XSCOQUSBVAKH</td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td><strong>Middle East (UAE)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/CNMHXHONGEWR</td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td><strong>Asia Pacific (Melbourne)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/FCCVJPUNPVXQ</td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td><strong>Europe (Zurich)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/IGPKWKCRTQSA</td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td><strong>Europe (Spain)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/IYZCFYKTLOEZ</td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/XSCOQUSBVAKH</td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td><strong>Middle East (UAE)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Control objective</strong></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/CNMHXHONGEWR</td>
</tr>
<tr>
<td></td>
<td><strong>Control API identifiers, by Region</strong></td>
<td></td>
<td><strong>Asia Pacific (Melbourne)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Framework</strong></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/FCCVJPUNPVXQ</td>
</tr>
</tbody>
</table>

**Control identifier**
- CT.CLOUDWATCH.PR.2

**Framework**
- NIST 800-53 Rev 5 AU-10
- NIST 800-53 Rev 5 AU-11

**Control objective**
- Establish logging and monitoring

**Control API identifiers, by Region**
- **Africa (Cape Town)**
  arn:aws:controltower:af-south-1::control/NSZIACKXCZ2WU
- **Middle East (Bahrain)**
  arn:aws:controltower:me-south-1::control/AMVXVAUSMISIHI
- **Israel (Tel Aviv)**
  arn:aws:controltower:il-central-1::control/UNHKBRUEDJVG
- **Europe (Zurich)**
  arn:aws:controltower:eu-central-2::control/IGPKWKCRTQSA
- **Europe (Spain)**
  arn:aws:controltower:eu-south-2::control/IYZCFYKTLOEZ
- **Asia Pacific (Hyderabad)**
  arn:aws:controltower:ap-south-2::control/XSCOQUSBVAKH
- **Middle East (UAE)**
  arn:aws:controltower:me-central-1::control/CNMHXHONGEWR
- **Asia Pacific (Melbourne)**
  arn:aws:controltower:ap-southeast-4::control/FCCVJPUNPVXQ
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
<td>east-2::control/KZLHFSHUXRWF</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ ZOBNLPOQHPDK</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-12</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ VPXBCGTGURFRG</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.7</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ HEBNGALBXYXJ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ QUYKUYVWSZV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ URQEHVTSKLLB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ TBWZGGFAGCX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ DNHPGCXFAJST</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ DSVHTVHFNRESE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ JPDUXRDODBEG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ TYKALNIDLUXK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ SBTRQDFYDCOK</td>
</tr>
</tbody>
</table>
| | | | Europe (Paris) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>west-3::control/QCSZCQYGUVUZ</td>
<td>• South America (São Paulo)</td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/ACYWMNWHXDKN</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/ACYWMNWHXDKN</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/ZWPLHRTNEWRN</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/FBRQUVHBOZPB</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/VNUHTQRCTWL</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/UZOKBNYSYIZW</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/OZCLFSHTIS</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/MDSOCYVCV</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/VEWMILFDETBA</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/EFQUCOH</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/DAUDMKIII</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/DPZXYQALXSB</td>
</tr>
</tbody>
</table>
Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
CT.CLOUDWATCH.PR.3 | • NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-3(6)  
• NIST 800-53 Rev 5 SC-12(2)  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-28  
• NIST 800-53 Rev 5 SC-28(1)  
• NIST 800-53 Rev 5 SC-7(10)  
• NIST 800-53 Rev 5 SI-7(6)  
• PCI DSS version 3.2.1 3.4 | • Encrypt data at rest | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
ZHBTZTMZYST  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
UEXYNAIAHTWI  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
RVGRWIMSJKPB  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
RZUDTGNQSLRG  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
YHKYBCWCRLLJ  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
AHEQLQIVIOWD  
• Europe (Frankfurt)  
arn:aws:controltower:eucentral-1::control/  
GSZUCKWWMLZM
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/GVVBWCAA0CTC</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/ZSQDFIHSSAOO</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/VZXNAZZKVVEJ</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/OTGQBNMWMNKV</td>
</tr>
<tr>
<td>(Mumbai)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/CWBBUMULYLOW</td>
</tr>
<tr>
<td>(Seoul)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/WYEDDRSFQTDY</td>
</tr>
<tr>
<td>(Tokyo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/WQCLCWXDFYLS</td>
</tr>
<tr>
<td>• South America</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/CXYHP0XPKX8WZ</td>
</tr>
<tr>
<td>(São Paulo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/CZEEVPANZGQK</td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/QAJHRGEGPDMX</td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/TGZQHFRKWOCF</td>
</tr>
<tr>
<td>(Jakarta)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.CLOUDWATCH.PR.4 | NIST 800-53 Rev 5 AU-6(1) | Establish logging and monitoring | • US East (N. Virginia) arn:aws:controltower:us-

northeast-3::control/ AZANGXXVSQGH
• Europe (Milan) arn:aws:controltower:eu-south-1::control/ ORYXUPSLDOOT
• Africa (Cape Town) arn:aws:controltower:af-south-1::control/ OLOFVUGRRHUK
• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/ ELINFWHYRXRZ
• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/ GMIMEPVNBZZ
• Europe (Zurich) arn:aws:controltower:eu-central-2::control/ TMUERIGFSBVJ
• Europe (Spain) arn:aws:controltower:eu-south-2::control/ SYOYMUBLETAN
• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/ DEJFOZEGREQP
• Middle East (UAE) arn:aws:controltower:me-central-1::control/ JBRQFIIYNSH
• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/ RFQBNRCCEIKA
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>

- **east-1::control/** UUBJSMHHQ8TE
- **US East (Ohio)**
  arn:aws:controltower:us-east-2::control/
  XFOXFWKOUFVI
- **US West (Oregon)**
  arn:aws:controltower:us-west-2::control/
  LLRWYTMFYUFL
- **Canada (Central)**
  arn:aws:controltower:ca-central-1::control/
  IOQUGWNHEDSO
- **Asia Pacific (Sydney)**
  arn:aws:controltower:ap-southeast-2::control/
  CLDCBWOSIAUL
- **Asia Pacific (Singapore)**
  arn:aws:controltower:ap-southeast-1::control/
  ISMLVCTXZGFY
- **Europe (Frankfurt)**
  arn:aws:controltower:eu-central-1::control/
  OOGEPDLPGDWEM
- **Europe (Ireland)**
  arn:aws:controltower:eu-west-1::control/
  APPYZPTNBLGO
- **Europe (London)**
  arn:aws:controltower:eu-west-2::control/
  XRRBPVFHMAFT
- **Europe (Stockholm)**
  arn:aws:controltower:eu-north-1::control/
  FKOIMKGDJP3BQ
- **Asia Pacific (Mumbai)**
  arn:aws:controltower:ap-south-1::control/
  IHHAKHDRLOGU
- **Asia Pacific (Seoul)**
  arn:aws:controltower:ap-northeast-2::control/
  CFAQBJOYCLFH
- **Asia Pacific (Tokyo)**
  arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeast-1::control/AEHEUFZDKJQD</td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/HPAIEIOARUEQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo) arn:aws:controltower:saeast-1::control/VCHSALJXEQI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/UQPZKKDGYZCH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/TIHAFDHKOGZJ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/VKHCDMGU00K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/YHLIYCNGUAA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/JULHFFNMHXU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/EKMHYJGVZCH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/YCKCTTEXWPSW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/OGCHPVU0WJ8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/UHOAJRVPLHA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.CODEBUILD.PR.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT.CODEBUILD.PR.1</td>
<td></td>
<td>Use strong</td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td>authentication</td>
<td></td>
</tr>
<tr>
<td>SA-3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td></td>
<td>6.4.4</td>
<td></td>
</tr>
<tr>
<td>8.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US East (N. Virginia)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US East (Ohio)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canada (Central)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US East (N. Virginia)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US East (Ohio)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canada (Central)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US East (N. Virginia)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US East (Ohio)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canada (Central)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/OZCLPKRVOVHV</td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/NCDYZQYYXIMF</td>
</tr>
<tr>
<td>Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/CZYKFOXD0XAE</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/LAPQRZRTCBBL</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/PMZ3KQNVBRKZ</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/XWOSM5CORYXY</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/EUUKRLJQRWDI</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/ LFQQPETWUEWP</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/PMRWCRVSHCSO</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/EHWDQPA0QVY</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/EJM3ERDNUJ8QJ</td>
</tr>
</tbody>
</table>
| Asia Pacific (Jakarta) |      |                  | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>southeast-3::control/LBGSSRBSSGUS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/QVLWIFOPPDEP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/MKNRUEOIFDIO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/WVYQZRGTLWHG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/POTJCWBCWTKI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/ODCFYBUSVHPI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/JPCHEZJSIYJY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/TLIZGYOSLXID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/BGONJAECQQODC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/YSLFINUPIWAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Melbourne)</td>
<td>arn:aws:controltower:ap-southeast-4::control/LXVUSLOKWAYD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.CODEBUILD.PR.2  | NIST 800-53 Rev 5 IA-5(7)  
 NIST 800-53 Rev 5 SA-3  
 PCI DSS version 3.2.1 8.2.1 | Use strong authentication | US East (N. Virginia)  
 arn:aws:controltower:us-east-1::control/  
 EJNHNFZRAMFG  
 US East (Ohio)  
 arn:aws:controltower:us-east-2::control/  
 QFRBXZNMGMLD  
 US West (Oregon)  
 arn:aws:controltower:us-west-2::control/  
 WWHTDFQNALJW  
 Canada (Central)  
 arn:aws:controltower:ca-central-1::control/  
 SCRYYBUZHBYQ  
 Asia Pacific (Sydney)  
 arn:aws:controltower:ap-southeast-2::control/  
 CDUPZGSWHYDK  
 Asia Pacific (Singapore)  
 arn:aws:controltower:ap-southeast-1::control/  
 FTDDOMZPCICH  
 Europe (Frankfurt)  
 arn:aws:controltower:eu-central-1::control/  
 DDVQLINSURBW  
 Europe (Ireland)  
 arn:aws:controltower:eu-west-1::control/  
 WEYMGAAEESOZ  
 Europe (London)  
 arn:aws:controltower:eu-west-2::control/  
 QZKDWOTRWSLC  
 Europe (Stockholm)  
 arn:aws:controltower:eu-north-1::control/  
 NACQKIAFRVFP  
 Asia Pacific (Mumbai)  
 arn:aws:controltower:ap-south-1::control/  
 MVSGVYHINMLZ |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/E00PASTMJSET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/DCWLYNEJULNR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/FHZHYYESCQDF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/SIANVCWRXGWJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/WTHWLRJZZOOW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/VYDLJEUYORHK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/JTEUUIMXIUPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/YPIAEPNIJNQG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/JEUDKXERRMYB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/GCGSHXWTTIVO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/BNEGCEZBVNSJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.CODEBUILD.PR.3  |  | • Establish logging and monitoring | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/YEHYWYAUIQHZ  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/UCDPZLB3EG6BZ  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/MJRSWBIBIW  
• Canada (Central)  
ar:n:aws:controltower:ca-central-1::control/ZNXNBZCHFPQ  
• Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/TYKATGNYMSLJ  
• Asia Pacific (Melbourne)  
ar:n:aws:controltower:ap-southeast-4::control/TYKATGNYMSLJ  
• Middle East (UAE)  
ar:n:aws:controltower:me-central-1::control/BXMYBACIZYGC  
• Asia Pacific (Hyderabad)  
ar:n:aws:controltower:ap-south-2::control/ZQLUVNOAMURG  
• Europe (Spain)  
ar:n:aws:controltower:eu-central-2::control/QGKVBMJEDJEW  
• Europe (Zurich)  
ar:n:aws:controltower:eu-central-1::control/MBMBIGGAVEJN  
<p>| NIST 800-53 Rev 5 AC-2(12) |  |  |  |
| NIST 800-53 Rev 5 AC-2(4) |  |  |  |
| NIST 800-53 Rev 5 AC-4(26) |  |  |  |
| NIST 800-53 Rev 5 AC-6(9) |  |  |  |
| NIST 800-53 Rev 5 AU-10 |  |  |  |
| NIST 800-53 Rev 5 AU-12 |  |  |  |
| NIST 800-53 Rev 5 AU-2 |  |  |  |
| NIST 800-53 Rev 5 AU-3 |  |  |  |
| NIST 800-53 Rev 5 AU-6(3) |  |  |  |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
<td>southeast-2::control/QYCIHTAXYLTZ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-9(7)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/VBDUMCPMLQTK</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/UTFBCSBISVW</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/EBCCRSBBQ0EK</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/LTEOHIIZTPFH</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/BGMCKVUVIAAXA</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/OWJXWKPQUCWT</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/BYCXOWVCFC0P</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/YFGOVO0KMZXQ</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/QNQUMYVIDFEX</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/KQRSGHFYHVUK</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/VEHXVCDHACPS</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/KOZNXUGUJOXA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/WKTXNANZYYL8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/MRUGSCUAOCA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/AQXBYQXZQESG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/DMZLXFFHATNX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-central-1::control/VUBGTVNAYWXX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/IFWAVZCWRNS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/CMIJY3CFEKWD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/DGQLWMRIBAOA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/KEFQCNMQBGJR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/CKDSLYLCJQRL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Asia Pacific (Melbourne) | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.CODEBUILD.PR.4</td>
<td></td>
<td></td>
<td>southeast-4::control/JeZZFrMLKwFS</td>
</tr>
</tbody>
</table>

CT.CODEBUILD.PR.4

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.CODEBUILD.PR.4</td>
<td>NIST 800-53 Rev 5 AC-2(1)</td>
<td>Enforce least privilege</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/RBxKqFzQEPsp</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/PTLLPjYJUzQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/DHCOAMLWMKyM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/0GBAKFmGJQzy</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-5</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/XWXSPghBLrFI</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/KHJHMUSOSACW</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6(10)</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/BINKIX3ANDN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6(2)</td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/DWJMNcZPnPQB</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1</td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/CLQMYJUR0D8</td>
</tr>
<tr>
<td></td>
<td>8.2.1</td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/MUHKwSBZQHhx</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective, by Region</td>
<td></td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------</td>
<td>------------------------------</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/RWTLUYFSMXLC</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/PBJCHJANZLVPL</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/JONXRDNZMTAS</td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/ZNKDUDDRGIYXPO</td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/DLHTVNXDNFNR</td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/NM0EJRZEUUSQ</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/LSZKDLJYXOEC</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/MCOFGENCDXMLZ</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/RANLWU0VCVOF</td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/VMQVAYCAMXDO</td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/FBQANITCHZIB</td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td>arn:aws:controltower:me-</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.CODEBUILD.PR.5  |           | Encrypt data at rest | US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/OJSTETVYFKHE  
US East (Ohio)  
arn:aws:controltower:us-east-2::control/XELBXZJFB6YK  
US West (Oregon)  
arn:aws:controltower:us-west-2::control/NTHPULVMASKY  
Canada (Central)  
arn:aws:controltower:ca-1716 |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>central-1::control/KJE0AYEPVAVT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/YKADOBILMOXE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/BRCVOCSEPXIQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/FPCUV5CXOALO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/HSYKLYVTHMSC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/WLFHTAHNSJRK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/KSTQQLUUMAB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ZUBAPDKBHPEE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/JIRHXXYRRCPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/QMYRAUAQSPG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/CUBUEWNCWWLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:saeast-1::control/EOQMRPVCYIVQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/X0OYSBYTMVEE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/IIQ0EVFKXQU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/BILMNUNVKGQU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/XIPONLRXVTLM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/EWFVDYKGEDQM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/YZAEEAYHVZTZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/RFR3CMVGRGLK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/PVMESLBTIORK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/EZEDZUYHE0ZT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/ETTGKYPZHDWE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/IKW5XRZWZGCL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.CODEBUILD.PR.6</td>
<td></td>
<td>Encrypt data at rest</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>• US East (N. Virginia)</td>
<td>arn:aws:controltower:us-east-1::control/WEEZOKNMKWFK</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3(6)</td>
<td>• US East (Ohio)</td>
<td>arn:aws:controltower:us-east-2::control/ARXGHZDJMVEV</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28</td>
<td>• Canada (Central)</td>
<td>arn:aws:controltower:ca-central-1::control/LWZACZNXEWFR</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28(1)</td>
<td>• Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/BKDPGCPXEHUYM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-7(6)</td>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/LJAFCKIMTDGT</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 10.5</td>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/LJXNLOULCHOC</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 10.5.2</td>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/RRRKAZWMMWC</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2</td>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/TUDOZOFBNLBJ</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/RWIXRMHPKSME
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/NKXXSNQNQJNAG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/XFEBDNHXUVOK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/MFNMFPARKBBF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/RXRLKZGUXBFQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/YLCNQJWSXLBS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/UFNWWPKHJXLL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/AWLHICDELWPN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/DQVTQLQFTTB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/RCUQGUPBRITK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/BJZEVERNBVKMT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/LKLEVOQIMCUV</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Africa (Cape Town) | arn:aws:controltower:af-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.DAX.PR.1        | • NIST 800-53 Rev 5 CA-9(1)  
|                    | • NIST 800-53 Rev 5 CM-3(6)  
|                    | • NIST 800-53 Rev 5 SC-13    
|                    | • NIST 800-53 Rev 5 SC-28    | • Encrypt data at rest  
|                    |                        | • US East (N. Virginia)  
|                    |                        | • US East (Ohio)          
|                    |                        | • US West (Oregon)        |
| south-1::control/  | US East (N. Virginia)  
| LXQDTRHEMKEN       | • Encrypt data at rest  
|                    | • US East (Ohio)        
|                    | • US West (Oregon)      |
|                   | US East (N. Virginia)  
|                   | • Encrypt data at rest  
|                   | • US East (Ohio)        
<p>|                   | • US West (Oregon)      |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td></td>
<td>west-2::control/LIDITLCMT8XF</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltOWER:ap-southeast-2::control/WQDLGHQPLEUEZ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltOWER:ap-southeast-1::control/QWQALHIYPDDD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltOWER:eu-central-1::control/USAYUGMVGVCZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltOWER:eu-west-1::control/HPQCLFNHNV5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltOWER:eu-west-2::control/YQFMNRBYQZUH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltOWER:ap-south-1::control/IPCTFYGTPJQK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltOWER:ap-northeast-1::control/YJPSKRTFOVNI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltOWER:eu-west-3::control/PBMMRKGQLQSP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltOWER:sa-east-1::control/XUCCOZBWAWSU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltOWER:us-west-1::control/OQJDQXHVVHYQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong) arn:aws:controltOWER:ap-east-1::control/YERSBEZNALIC</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/MAULWZOVBHAG</td>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/MAULWZOVBHAG</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/AVRDIKUBVIEL</td>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/AVRDIKUBVIEL</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eusouth-1::control/AHWDGTQANTPZ</td>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eusouth-1::control/AHWDGTQANTPZ</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/RKQETEUGOLZT</td>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/RKQETEUGOLZT</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:mesouth-1::control/TJAYEVJIWMC</td>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:mesouth-1::control/TJAYEVJIWMC</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/MEQGYHFDONJQ</td>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/MEQGYHFDONJQ</td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eucentral-2::control/WERBGVNLRWCP</td>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eucentral-2::control/WERBGVNLRWCP</td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eusouth-2::control/XTNHYZREDREV</td>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eusouth-2::control/XTNHYZREDREV</td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:apsouth-2::control/HDSPSOQHOADH</td>
<td>• Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:apsouth-2::control/HDSPSOQHOADH</td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/DNVHZNIVNVL</td>
<td>• Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/DNVHZNIVNVL</td>
</tr>
<tr>
<td>Asia Pacific (Melbourne)</td>
<td>arn:aws:controltower:apsoutheast-4::control/RYLGGCGVEELG</td>
<td>• Asia Pacific (Melbourne)</td>
<td>arn:aws:controltower:apsoutheast-4::control/RYLGGCGVEELG</td>
</tr>
</tbody>
</table>
## CT.DAX.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.DAX.PR.2        | • NIST 800-53 Rev 5 CP-10  
                     • NIST 800-53 Rev 5 CP-2(2)  
                     • NIST 800-53 Rev 5 CP-6(2)  
                     • NIST 800-53 Rev 5 SC-36  
                     • NIST 800-53 Rev 5 SC-5(2)  
                     • NIST 800-53 Rev 5 SI-13(5) | • Improve resiliency  
                     • Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
SBKTBSUMMEK  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
XOLXwDXOTSdS  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
BXXBFHSEQVc  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
JUNTGFKWFFJA  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
YLPPVNAcLdZ  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
LSYCOPMQSAqK  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/  
WYPOTMKgGRTf  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/  
GTNFGAXVMTIY  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/  
UDVRXZWHEH3LQ  
• Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/  
LKWsO1JNZRRt  
• Asia Pacific (Mumbai)  
arn:aws:controltower:ap-south-1::control/  
KIADSEVTATnk  

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ MCGGZQJDQSEG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ KCZJTMRTLPZH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ RDIXQORVUYTX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ WKKCRCYVVOYA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ HCQNVBKSJPXK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ GPIACDHSJUVO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ SDXGBJAMZIPN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ HGXHJVRQSTJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/ UNTRGVZHGCW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/ SILHNRKXURR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/ QVZKAFUMHFI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.DMS.PR.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **NIST 800-53 Rev 5 AC-21**
- **NIST 800-53 Rev 5 AC-3**
- **NIST 800-53 Rev 5 AC-3(7)**
- **NIST 800-53 Rev 5 AC-4**
- **NIST 800-53 Rev 5 AC-4(21)**
- **NIST 800-53 Rev 5 AC-6**
- **NIST 800-53 Rev 5 SC-7**
- **NIST 800-53 Rev 5 SC-7(11)**
- **NIST 800-53 Rev 5 SC-7(16)**  
  - Limit network access  
  - Enforce least privilege  
  - US East (N. Virginia)  
    - arn:aws:controltower:us-east-1::control/  
      T2MIGTNHDWQL  
  - US East (Ohio)  
    - arn:aws:controltower:us-east-2::control/  
      VSDQWVQWVEUVH  
  - US West (Oregon)  
    - arn:aws:controltower:us-west-2::control/  
      TEUGLCCMRFBE  
  - Canada (Central)  
    - arn:aws:controltower:ca-central-1::control/  
      XXXIGZQSPQV  
  - Asia Pacific (Sydney)  
    - arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(20)</td>
<td></td>
<td>southeast-2::control/ZERDIGHXMHK</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(21)</td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(3)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/FVTCTPWHDTL</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(4)</td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(9)</td>
<td></td>
<td>arn:aws:controltower:europe-central-1::control/OMLKCPQPSJTR</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.2.1</td>
<td></td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3</td>
<td></td>
<td>arn:aws:controltower:europe-west-1::control/YCHPEYBXVBF</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.1</td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.2</td>
<td></td>
<td>arn:aws:controltower:europe-west-2::control/WGO0QPOQGTCC</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.4</td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.6</td>
<td></td>
<td>arn:aws:controltower:europe-north-1::control/LEXIGWTCMJSAA</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>2.2.2</td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/UICXSEVCABSW</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.2</td>
<td></td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.4</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/PMXDPOCYXVY</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.6</td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>2.2.2</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/CTFMR3PUPGCC</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3</td>
<td></td>
<td>Europe (Paris)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.2</td>
<td></td>
<td>arn:aws:controltower:europe-west-3::control/EOSDYNKIGAWA</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.4</td>
<td></td>
<td>South America (São Paulo)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>1.3.6</td>
<td></td>
<td>arn:aws:controltower:saeast-1::control/XOKBSYBQKJOK</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>2.2.2</td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/TSMFXMFCPLVU</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/QAPCPDZHYTEBF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/CXVE0OHIMNNL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/CPG0CDMDHGF1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/OPSWXXYSYUMY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/UNWEF5SWLQTO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/ZWSGRHYTXZSN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/HZSULTIXLIGQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/UMVTLCRR9GKV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/TGGYQCVRRDGF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/EUCLHHUYDEEC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/LT5KNULFPUTG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Asia Pacific (Melbourne) | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.DOCUMENTDB.PR.1</td>
<td></td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/THVIKWXJQCEB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/AGSEOYNXOSRK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/HXVOHNCIJRUC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/CUUSQEVMNYIR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/LVZGOWHNFFQQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/XUVXHBPXAAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eucentral-1::control/XOBYNMAVPBF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/UWYKAUHSRUKZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/CFNPFPZENDE0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/CMAILYDYAUOY</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.DOCUMENTDB.PR.1</td>
<td></td>
<td></td>
<td>southeast-4::control/ROMIMQWYAGFD</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective, by Region</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>----------------------------</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/NJTDQSHRSPIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/TEZXETAGKYO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/EONQXNXGAPB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ACFERMCUQLYD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/NSSWQWRWJNFM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/TRACDDWLKDRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/NDCSWMCFZFWK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/BMNUBLZWWWA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/FIHFPXXSWPVR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/DTKBFEIQNQL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/AIQLWAENCEVZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.DOCUMENTDB.PR.2 |           | Improve resiliency | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/ARZAKWMXVIVC  
|                    |           |                  | • US East (Ohio) arn:aws:controltower:us-east-2::control/ZIBRJZXDMYK  
|                    |           |                  | • US West (Oregon) arn:aws:controltower:us-west-2::control/XIHJTNHDRQAF  
|                    |           |                  | • Canada (Central) arn:aws:controltower:can- 

---

CT.DOCUMENTDB.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.DOCUMENTDB.PR.2 |           | Improve resiliency | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/ARZAKWMXVIVC  
|                    |           |                  | • US East (Ohio) arn:aws:controltower:us-east-2::control/ZIBRJZXDMYK  
|                    |           |                  | • US West (Oregon) arn:aws:controltower:us-west-2::control/XIHJTNHDRQAF  
|                    |           |                  | • Canada (Central) arn:aws:controltower:can- 

---

AWS Control Tower User Guide

CT.DOCUMENTDB.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| south-1::control/GVXHDHTISRP |           |                  | • Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/ZKHDGAAGCYFF  
|                                  |           |                  | • Europe (Zurich) arn:aws:controltower:eu-central-2::control/UFMCCFRSISYY  
|                                  |           |                  | • Europe (Spain) arn:aws:controltower:eu-south-2::control/AIUTNQWKGFD  
|                                  |           |                  | • Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/ECVIEFMUQJMZ  
|                                  |           |                  | • Middle East (UAE) arn:aws:controltower:me-central-1::control/BBACQBVBH5DC  
|                                  |           |                  | • Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/KNYFLVIZ0IAZ  

---

AWS Control Tower User Guide

CT.DOCUMENTDB.PR.2
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>central-1::control/NURHPDKMITF</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td>central-1::control/</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>central-1::control/</td>
<td></td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td>central-1::control/</td>
<td></td>
<td></td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td>central-1::control/</td>
<td></td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td>central-1::control/</td>
<td></td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td>central-1::control/</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>central-1::control/</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>central-1::control/</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>central-1::control/</td>
<td></td>
<td></td>
<td>Europe (Paris)</td>
</tr>
<tr>
<td>central-1::control/</td>
<td></td>
<td></td>
<td>South America (São Paulo)</td>
</tr>
<tr>
<td>central-1::control/</td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/VTWVHTWZJXDS</td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/HTINNLMXRAD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-southeast-3::control/NTKVAPLDGPIOE</td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-northeast-3::control/TYKSQTKVDRPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-northeast-3::control/TYKSQTKVDRPU</td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/XDBAEAOQGFZQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:me-south-1::control/LBGOTUKVIBIY</td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-central-1::control/ZBVKSKSCLRSO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:il-central-1::control/TLEHFGYBIPPV</td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/TLEHFGYBIPPV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-central-2::control/YMMXEICWPZQC</td>
<td>• Europe (Zurich) arn:aws:controltower:eu-south-2::control/NOKRKREQWEPP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-central-2::control/YMMXEICWPZQC</td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/NOKRKREQWEPP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-south-2::control/WAJSWJSCJGCH</td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/WAJSWJSCJGCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:me-central-1::control/ZBVKSCLRSO</td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/ZBVKSCLRSO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CT.DYNAMODB.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.DYNAMODB.PR.1</td>
<td>• NIST 800-53 Rev 5 CP-10</td>
<td>• Improve resiliency</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/OIZVLCBZARKX</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CP-9</td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/MMOPKUVAJYZD</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-12</td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/FEOCXMTUXONR</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.1</td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td>• Improve resiliency</td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/MUUOXHEGOERO</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.1</td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td>• Improve resiliency</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/HYEXGAE0DQCQ</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.1</td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/BQIMDWJDEKLB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/PUHTGIFRKSNJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/EZQDFKIPCML</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/NWUTQTYYDZBH</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/CYGILLICLUBJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/RMGFYQOKBYAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/SICGAIHKEQSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/YLNVMSVQPPW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ZZZATCBZSQZS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/OWRFVRAEHORK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/AOEAJRZQUOJU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/IGOUAMZMLPQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/XAWHUFNRUXO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/CRICQPHNTAAAY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/CXPURYMVYMSC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.DYNAMODB.PR.2</td>
<td></td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/AVPWTCQJZVFL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/GLNZIQQIDTES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-1::control/KL58N3G3K42I</td>
</tr>
<tr>
<td>CT.DYNAMODB.PR.2</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/AVPWTCQJZVFL</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/GLNZIQQIDTES</td>
</tr>
</tbody>
</table>

- Middle East (Bahrain) arn:aws:controltower:me-south-1::control/ZREDRAAHQUB
- Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/IKHNTWYSGUQ
- Europe (Zurich) arn:aws:controltower:eu-central-2::control/DMUBHZWCMKBV
- Europe (Spain) arn:aws:controltower:eu-south-2::control/EFNQESSMTRWA
- Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/CUNOZJCLCDXS
- Middle East (UAE) arn:aws:controltower:me-central-1::control/UPBXICUCGOPG
- Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/HOSCJVRPLNZP
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>west-2::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td>RIOHZBTKOBKP</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td>ZXANARWCPQNWW</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td>MXJNGHQYSQWT</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td>VATDYITSDFZT</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td>CBZJAP2HXDFD</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td>VNDWOJRWDJIZ</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td>RIBSFVFNIAR</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td>IADWVKAFGXUW</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td>WZFVBOYAOIMH</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td>PHGDRCMDDFER</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td>CREBIEGAAVZS</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td>QVMDXANQFBCA</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>arn:aws:controltower:sa-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>LZFDTLNKOQXX</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td>HMMSN0FBEK</td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td>arn:aws:controltower:us-</td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td>west-1::control/</td>
<td></td>
<td></td>
<td>HMMSN0FBEK</td>
</tr>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>PRMTFXGWHHGWM</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>PRMTFXGWHHGWM</td>
</tr>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td>PCYSWLZHMHPW</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td>northeast-3::control/</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td>QFZXLJQCUPTB</td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td>arn:aws:controltower:eusouth-1::control/</td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td>GCGXGJULM9J</td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td>arn:aws:controltower:af-</td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td>south-1::control/</td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td>FWOEVCM9ZRIE</td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td>(Bahrain)</td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td>arn:aws:controltower:mesouth-1::control/</td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td>PBRYEFBWDTNV</td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td>arn:aws:controltower:il-</td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td>central-1::control/</td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td>SWUAOFXQZSO</td>
<td></td>
<td></td>
<td>• European (Zurich)</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td>arn:aws:controltower:eusouth-central-2::control/</td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td>TRYHWAUMG53</td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td>arn:aws:controltower:eusouth-central-2::control/</td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td>GFIJ9UMQZ5YKT</td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td>(Hyderabad)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td>south-2::control/</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td>MOYLIDKDFBHL</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
</tbody>
</table>
### CT.EC2.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.EC2.PR.1        | NIST 800-53 Rev 5 AC-21 | • Enforce least privilege  
• Protect configurations | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/NOTMRPTQRRAA  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/SMAKVTVP3EPX0  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/HAYOFUSRHZIZ  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/HHAHBCLQXYX  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/1ZBFO3RZKBEV  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/UKDFSIYPFVM  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/ADABIOVNNTMS  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/GFKNBAOPNXHF |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/WYVSVHJYIHAD</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/UTXKBQQERGVI</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/OZFCKKRKZSIB</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/PRPQIPYUCLFE</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/YSQCFBU0TEPD</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2</td>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/YAIRLGKSVPZS</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/IJAIPRTDFHRB</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/QEIYONGZGZCVI</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/PNKPPQYMTJFF</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/PWXEQPXBWGT</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/RDUEMSJPXBWR</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-west-3::control/1740</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.EC2.PR.10</td>
<td>None</td>
<td>• Establish logging and monitoring</td>
<td>• US East (N. Virginia) Arn:aws:controltower:us-east-1::control/ VAWEVAFBTTYX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) Arn:aws:controltower:us-east-2::control/</td>
</tr>
</tbody>
</table>

### CT.EC2.PR.10

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>south-1::control/</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:af-south-1::control/ RDGRKFSEPUCK</td>
</tr>
<tr>
<td>WLUDDFWBDJHU</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:me-south-1::control/ EGMHACFCYINC</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:il-central-1::control/ FUBCShOJvema</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:eu-central-2::control/ IMRVGYYQJLLW</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>Arn:aws:controltower:me-south-1::control/ EGMHACFCYINC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:eu-south-2::control/ VNHGwWCSJWRA</td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>Arn:aws:controltower:eu-central-2::control/ IMRVGYYQJLLW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-south-2::control/ MTSLSKQKPIFR</td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>Arn:aws:controltower:ap-south-2::control/ MTSLSKQKPIFR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:me-central-1::control/ FEMJIFSFEJ0I</td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>Arn:aws:controltower:me-central-1::control/ FEMJIFSFEJ0I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-southeast-4::control/ SPYIFBBRIAND</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>east-2::control/</td>
<td></td>
<td></td>
<td>ZXUWTNZZOQUI</td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/MPIXENDLDJMR</td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/RTCRBMNLMBL</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/PBNLHCFGRZZO</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/XXRXMNFPAMIR</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/HPOMPECFAIJ</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/DBKJHDJXKWNY</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/CTADRPGQIHX</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/LAUHKTATLXY</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/IUSDZWDETIR</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/GNMPIMBYVNYKE</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/PRVLLJPYMYPY</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>west-3::control/</td>
<td></td>
<td></td>
<td>YJMLRLGRBSE</td>
</tr>
<tr>
<td>• South America</td>
<td></td>
<td></td>
<td>(São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VYTXXSGZXJJA</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td>California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>YCL3PFXTXKZYYU</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HISQJLESBWSD</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GNBOOTBHJQVG</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GBITMKPAKCE</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DNLPNFANEMRU</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ZBXPGHGLMNXH</td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td></td>
<td>(Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KDLJUDSYZQNF</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TJMFQWKSYRGJ</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eucentral-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EEVRMJJQGH1FB</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eusouth-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DCORPIAUVCG</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------</td>
<td>-----------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.EC2.PR.11</td>
<td></td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/USZSBVLKBFJW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/GAJQJEHKYENL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/NMURDTFOJDZH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ZBwKQBJUYPUA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/HNTCZVIEEJOI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/GAEZXFNKITWD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/PDKCUGSWC3WF</td>
</tr>
</tbody>
</table>

CT.EC2.PR.11

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.EC2.PR.11</td>
<td></td>
<td>• Limit network access</td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/TWBJIMNVOQYC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/XCQTHTPUPGSIF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/LYBOKWDBULNW</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 AC-21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/RAMFRLQUVVMN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/XRFWIPMMYBWF</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/TXQCCBSQPFNB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/IRKWHFEJOBYB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/WOPJHTRSHEFP</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ENUFAUVEZVFI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/AZQORBOHZJMI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/BCNLWTEEHTHE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/TLXGBVBTJUVZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/GROFHQVWGNJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/CRAITTP0FXYCU</td>
</tr>
</tbody>
</table>
| | | | • Asia Pacific (Osaka) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.EC2.PR.12       | NIST 800-53 Rev 5 AC-4(21) | Limit network access | US East (N. Virginia) arn:aws:controltower:us-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>east-1::control/QNDKUCEIGRNB</td>
</tr>
<tr>
<td>• US East (Ohio)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/IFRKPJEDMUU</td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/VHEHBTESLGUN</td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/HJXXVHTKUFBC</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/DRPWGXJHMIQA</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/INYDINMLKHFD</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/EBSIWJJPVVBW</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/SDYXHOVBMM</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/JWYSLVOEA0JJ</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/SIJDBTELNSUI</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/VEHFJHVKNR</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/HFYUZFTNQRZG</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo) | | | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeast-1::control/YBIIKCGWAORD</td>
<td></td>
<td></td>
<td>Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/JCOYUSUWIRJX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/UDKUYKZVSBVJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/WJSMEAZXXXVG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/LSZFXIAPRTNL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/UQOCEESBNIF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/DRLFKVUWERWU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/EPDEAPQFUAPZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/ERDLIQVOFJU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/PTDARPFBEGAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/TGQZBFRIVBZD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/VP0JZQAASXUU</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.EC2.PR.13</td>
<td>NIST 800-53 Rev 5 CA-7</td>
<td>Establish logging and monitoring</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ FZSUCEEYAPWI</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ ANUKCTRRCCEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ BWUUBNNIBGXB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ MTUNNERTAWUK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ O5VHEFPIYSYR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ Q2MARZRDZENJ</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/ MYCUMPVJEXFZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/ QCBNFLLWHORL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ ZJDKEANFYUMV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ ARMDLUGJAMIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ SSCKAJIEWDZ0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ OACEJPSHEDTT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ OHYRLAGKXSXR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ BWHESSZGTPFW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ HLGYBXIMQRHU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ BTLZSJHIZWG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ DTDHLEWFSGSKV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>southeast-3::control/JYEUKJNOFJQT</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ZBKKUULMNWGL</td>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/AYFZQOECAPETH</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/QQXKJMHCUFTX</td>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/YOPBAHWMVOA</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/YOPBAHWMVOA</td>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/YCZVKPCTQO0V</td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/0OEREZRXCQHBCZ</td>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/SZRCWAQMISFS</td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/LXNQLJAJMVL</td>
<td>Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/EWZWVABCT0CS</td>
</tr>
<tr>
<td>Asia Pacific (Melbourne)</td>
<td>arn:aws:controltower:ap-southeast-4::control/YXZZGZGJVL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CT.EC2.PR.14

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.EC2.PR.14</td>
<td></td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/MQKNFAX3GETK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/VMCVOKHAAOFHC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/AGCVMUH0LSMD</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Encrypt data at rest</td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/CGSRRXAEKBDN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3(6)</td>
<td>Encrypt data at rest</td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/GRIQSOWFUTUZ0</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-13</td>
<td>Encrypt data at rest</td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/GIIXBVQAQTNT</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28</td>
<td>Encrypt data at rest</td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/RIERQ3RBCV5</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28(1)</td>
<td>Encrypt data at rest</td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/VBAWHSRPRVFO</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(10)</td>
<td>Encrypt data at rest</td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/TBTEUVAWFLJG</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-7(6)</td>
<td>Encrypt data at rest</td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/KFIXEWMHUEYPE</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2</td>
<td>Encrypt data at rest</td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/TUQFRYIKJBAM</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 3.4</td>
<td>Encrypt data at rest</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 8.2.1</td>
<td>Encrypt data at rest</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/WCOMSKMDNAFG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/LUUPJIKOPJWO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/QTNFAUGUTMHB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/CCWGUXJARQL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/PHUZHFOQIHAY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/YGASEHTTCVJZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/VMKDWYCKPRH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/IZA2FFYMCGZOS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/IALXISTUWWM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/PPwMLTWEAVHD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/TKROLDYAXJKI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CT.EC2.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Enforce least privilege&lt;br&gt;• Protect configurations</td>
<td>US East (N. Virginia)&lt;br&gt;arn:aws:controltower:us-east-1::control/CFVETGC3NKXF&lt;br&gt;US East (Ohio)&lt;br&gt;arn:aws:controltower:us-east-2::control/HFMY00XSTPVC&lt;br&gt;US West (Oregon)&lt;br&gt;arn:aws:controltower:us-west-2::control/GQRIEQC050B</td>
</tr>
<tr>
<td>CT.EC2.PR.2</td>
<td>NIST 800-53 Rev 5 CM-2</td>
<td>• Protect configurations</td>
<td>Canada (Central)&lt;br&gt;arn:aws:controltower:can-central-1::control/KLTLENUYLCHX&lt;br&gt;Asia Pacific (Sydney)&lt;br&gt;arn:aws:controltower:ap-southeast-2::control/NAGYSJVVWTD5</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2(2)</td>
<td>• Protect configurations</td>
<td>Malaysia (Singapore)&lt;br&gt;arn:aws:controltower:ap-southeast-3::control/TREGJKJNQ15</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.1.1</td>
<td>• Protect configurations</td>
<td>Asia Pacific (Melbourne)&lt;br&gt;arn:aws:controltower:ap-southeast-4::control/PTTAGSYATZTH</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td>southeast-2::control/TPIXAIDPGBY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/TTTKTHOMEJGW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/LDZLIZEYHSBD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/XOCHLPP00VT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/CJCTXIIMNFJX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/YYGFVEDLRYTJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/UMDAUNXLFJJI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/UELALJZTKJG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/IQTZLHSDAVPJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/SSTRQDHRJFJX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/AHSIJDKCCKEVD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/FLIITHQMTLZG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/OXDELGPZPGWI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/NSYQYXIHEBWE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/UWVOMYBTMINT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/TPMKJPKVEDNA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/DUKUFDMKZNNE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/RBSTYRGNTGHQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/FRMAUAMPFCPJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/CJVAWZZZEAFY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/VHBUBKOEZTHI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/AKETHXXMVTQR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/EHLAXLXRJAQD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Melbourne)</td>
<td>arn:aws:controltower:ap-1756</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## CT.EC2.PR.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.EC2.PR.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **NIST 800-53 Rev 5 AC-4**
- **NIST 800-53 Rev 5 AC-4(21)**
- **NIST 800-53 Rev 5 SC-7**
- **NIST 800-53 Rev 5 SC-7(11)**
- **NIST 800-53 Rev 5 SC-7(16)**
- **NIST 800-53 Rev 5 SC-7(21)**
- **NIST 800-53 Rev 5 SC-7(4)**
- **NIST 800-53 Rev 5 SC-7(5)**
- **PCI DSS version 3.2.1 1.2.1**
- **PCI DSS version 3.2.1 1.3**
- **PCI DSS version 3.2.1 1.3.1**
- **PCI DSS version 3.2.1 1.3.2**
- **PCI DSS version 3.2.1 1.3.4**
- **PCI DSS version 3.2.1 1.3.6**
- **PCI DSS version 3.2.1 2.2.2**
- **Limit network access**
- **Enforce least privilege**

- **US East (N. Virginia)**
  - `arn:aws:controltower:us-east-1::control/ VVDNPMPRRUW/KZ`
- **US East (Ohio)**
  - `arn:aws:controltower:us-east-2::control/ PZRWTPBHWZVJ`
- **US West (Oregon)**
  - `arn:aws:controltower:us-west-2::control/ ZDDEIIIPHTMLA`
- **Canada (Central)**
  - `arn:aws:controltower:can-central-1::control/ XENHMLFNKUIK`
- **Asia Pacific (Sydney)**
  - `arn:aws:controltower:ap-southeast-2::control/ WUNOAUPAPJBR`
- **Asia Pacific (Singapore)**
  - `arn:aws:controltower:ap-southeast-1::control/ LZMEYSKSQYSC`
- **Europe (Frankfurt)**
  - `arn:aws:controltower:eu-central-1::control/ TKGSLZFXKVYVW`
- **Europe (Ireland)**
  - `arn:aws:controltower:eu-west-1::control/ KEAWNIFQNOQOD`
- **Europe (London)**
  - `arn:aws:controltower:eu-west-2::control/ RLLXYAJVITSR`
- **Europe (Stockholm)**
  - `arn:aws:controltower:eu-north-1::control/ RASRTLWUAOU`
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>• Asia Pacific (Mumbai)</td>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/QIMWICZWNCTJ</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>• Asia Pacific (Seoul)</td>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/TLXGQRQZRNJR</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>• Asia Pacific (Tokyo)</td>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/DCJBSGYTZDVE</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>• Europe (Paris)</td>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/OOHNEEZAASSE</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>• South America (São Paulo)</td>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/QAMRZHHGXINM</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>• US West (N. California)</td>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/LBDDZFPUBGSA</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>• Asia Pacific (Hong Kong)</td>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/WOHIMVFOWTAL</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>• Asia Pacific (Jakarta)</td>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/VCSBDJNKNXJR</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>• Asia Pacific (Osaka)</td>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/VFVJCOQMJKBH</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>• Europe (Milan)</td>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/IXEOZHLDVITA</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>• Africa (Cape Town)</td>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/YHTHJBQJJVFQ</td>
</tr>
</tbody>
</table>
| Middle East (Bahrain) | • Middle East (Bahrain) | • Middle East (Bahrain) | arn:aws:controltower:me-
### CT.EC2.PR.4

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td>central-1::control/KPFAQFAQJAWK</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Asia Pacific (Sydney)**
  - arn:aws:controltower:ap-southeast-2::control/GEQABOPVFXCO
- **Asia Pacific (Singapore)**
  - arn:aws:controltower:ap-southeast-1::control/BWPBQRUMLZXK
- **Europe (Frankfurt)**
  - arn:aws:controltower:eu-central-1::control/NUMPUVXQDMGA
- **Europe (Ireland)**
  - arn:aws:controltower:eu-west-1::control/NMIWXPPBIIJC
- **Europe (London)**
  - arn:aws:controltower:eu-west-2::control/BMSDUHXWMANZ
- **Europe (Stockholm)**
  - arn:aws:controltower:eu-north-1::control/CZTDWXPXRIDER
- **Asia Pacific (Mumbai)**
  - arn:aws:controltower:ap-south-1::control/VALJQOMVXOMI
- **Asia Pacific (Seoul)**
  - arn:aws:controltower:ap-northeast-2::control/YTLKASXYASWH
- **Asia Pacific (Tokyo)**
  - arn:aws:controltower:ap-northeast-1::control/MHXZHOBFJFUN
- **Europe (Paris)**
  - arn:aws:controltower:eu-west-3::control/AJJMZXCYCZHP0
- **South America (São Paulo)**
  - arn:aws:controltower:sa-east-1::control/JZHLGQYTWPNK
- **US West (N. California)**
  -
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>arn:aws:controltower:us-west-1::control/PVHDAKICIBKQU</td>
<td><strong>Asia Pacific (Hong Kong)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-east-1::control/JYOPIABWXZK0</td>
<td><strong>Asia Pacific (Jakarta)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-southeast-3::control/ERCKYPEQKQY</td>
<td><strong>Asia Pacific (Osaka)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-northeast-3::control/BNIHBVSGLUNE</td>
<td><strong>Europe (Milan)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-south-1::control/GFGHGDGRWXGM</td>
<td><strong>Africa (Cape Town)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-south-1::control/NLXKLURXRNDL</td>
<td><strong>Middle East (Bahrain)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:me-south-1::control/NUDOLFIXZEYZ</td>
<td><strong>Israel (Tel Aviv)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:il-central-1::control/FDAHWSZJUBGN</td>
<td><strong>Europe (Zurich)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eucentral-2::control/VBQSHMYYHTM</td>
<td><strong>Europe (Spain)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eusouth-2::control/DRDSQMCKJR</td>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-south-2::control/KMFIMHJPR</td>
<td><strong>Middle East (UAE)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:me-central-1::control/NXZKINMFEMOI</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CWCT.EC2.PR.5

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.EC2.PR.5        | • CIS AWS Benchmark 1.4 5.1 | • Limit network access | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
DUQKORYIWJYQ  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
OEYVCMLVLVIAT  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
JZWYQ0ZYBXKL  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
FTWNDUKURLTP  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
JSPEJFTBYZEF  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
XTGIABCSEGPH  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/  
SWBFL3HBRLPW  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/  
GNLTIWRBKZGMV  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/  
XMREZWZEQWZR  

- Asia Pacific (Melbourne)  
arn:aws:controltower:ap-southeast-4::control/  
LPEKGIJIK0SHF  
- NIST 800-53 Rev 5  
  AC-4(21)  
- NIST 800-53 Rev 5  
  CA-9(1)  
- NIST 800-53 Rev 5  
  CM-2  
- NIST 800-53 Rev 5  
  CM-2(2)  
- NIST 800-53 Rev 5  
  CM-7  
- NIST 800-53 Rev 5  
  SC-7  
- NIST 800-53 Rev 5  
  SC-7(21)  
- NIST 800-53 Rev 5  
  SC-7(5)  
- PCI DSS version 3.2.1  
  1.2.1  
- PCI DSS version 3.2.1  
  1.3  
- PCI DSS version 3.2.1  
  1.3.1  
- PCI DSS version 3.2.1  
  1.3.2  
- PCI DSS version 3.2.1  
  1.3.4  
- PCI DSS version 3.2.1  
  2.2  
- PCI DSS version 3.2.1  
  2.2.2
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/EJCXPRMBYXHH</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/PEKEYNXLWHA</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/LMVUECWMAG</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/BWLZVKGFDRIG</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/GCHXILWGOPBI</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/LMJNAXCWFKLK</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/GONSQXTKTNDT</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/NXUWUXUAOGPAS</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/XNKLOVHBEZ</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/HPUAFLSJFNNQ</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/LSSLWVUITJC</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.EC2.PR.6        | NIST 800-53 Rev 5 AC-4(21)  
NIST 800-53 Rev 5 CA-9(1)  
NIST 800-53 Rev 5 CM-2  
PCI DSS version 3.2.1 1.2.1 | Limit network access | US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/THHTKPOPPVGC  
US East (Ohio)  
arn:aws:controltower:us-east-2::control/FAQMZVDILTRV  
US West (Oregon)  
ar#:aws:controltower:us--
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>west-2::control/FSNQGGGGOUBAE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/DKUIAHCHYZTU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/NCDQQZLDCZI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/TJWCZQYWQVXKW</td>
</tr>
<tr>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/VGTXIUNUITGO</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-west-1::control/KVVDNFFOMGMT</td>
</tr>
<tr>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/KVVDNFFOMGMT</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-2::control/XAXENCPCHDJK</td>
</tr>
<tr>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/TIUYQFRWXSBR</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-west-3::control/UKGTSWXII0EJ</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/RSPKEHRKFSKK</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-northeast-2::control/UBNOJKELNWGQ</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-1::control/XJFUTUKMIXHKB</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-1::control/XJFUTUKMIXHKB</td>
</tr>
<tr>
<td>• South America (São Paulo) arn:aws:controltower:sa-</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>FWVYDJQAPZUF</td>
</tr>
<tr>
<td>US West (N.</td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/AXCDWYYFELJJ</td>
</tr>
<tr>
<td>Asia Pacific (Hong</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td>Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/JGBJBNJYKXY5</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/OEHVBKEUNBFI</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/RKIVCDUNAHHS</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/YPBJPJCBEKRXG</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/VFXWQXULUIHI</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/DQKFUMTNBEXM</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/SGAIFIRXPCUH</td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/TSZFSULSYZVH</td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/YCUDIIRUMWEV</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td>(Hyderabad)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/PQKQBDQPIXCA</td>
</tr>
</tbody>
</table>
## CT.EC2.PR.7

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.EC2.PR.7        |           | Encrypt data at rest | US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/TORIRIQKVTC5  
US East (Ohio)  
arn:aws:controltower:us-east-2::control/KWFITNUSNTWO  
US West (Oregon)  
arn:aws:controltower:us-west-2::control/YJXXNEDHNQGK  
Canada (Central)  
arn:aws:controltower:ca-central-1::control/KMLZLRNMCKXY  
Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/QVYBOWBJXKC  
Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/FSXHMDVXAKTQ  
Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/NILBBCJYSTMS  
Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/CZSRNXVWVWON  |

- Middle East (UAE)  
  arn:aws:controltower:me-central-1::control/UKUCDQFEDMDGT  
- Asia Pacific (Melbourne)  
  arn:aws:controltower:ap-southeast-4::control/ATKGMNOIYDJQ  
- NIST 800-53 Rev 5 CA-9(1)  
- NIST 800-53 Rev 5 CM-3(6)  
- NIST 800-53 Rev 5 SC-13  
- NIST 800-53 Rev 5 SC-28  
- NIST 800-53 Rev 5 SC-28(1)  
- NIST 800-53 Rev 5 SC-7(10)  
- NIST 800-53 Rev 5 SI-7(6)  
- PCI DSS version 3.2.1  
  2.2  
- PCI DSS version 3.2.1  
  3.4  
- PCI DSS version 3.2.1  
  8.2.1  

1767
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (London)</td>
<td>Arn:aws:controltower:eu-west-2::control/ JRGGEILMPZCB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>Arn:aws:controltower:eu-north-1::control/ SKZQW0OFKNWA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>Arn:aws:controltower:ap-south-1::control/ MXXCFQMDZVYY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>Arn:aws:controltower:ap-northeast-2::control/ XMLXJMIJGLVL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>Arn:aws:controltower:ap-northeast-1::control/ ZAXKMESMEXSF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>Arn:aws:controltower:eu-west-3::control/ ZQJJBP0ZJEJK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>Arn:aws:controltower:sa-east-1::control/ HMVXESEHHIEFX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>Arn:aws:controltower:us-west-1::control/ PQKLU3PAIDPH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>Arn:aws:controltower:ap-east-1::control/ NIMAUKZKMOIE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>Arn:aws:controltower:ap-southeast-3::control/ ZPFLDJXNZPIE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>Arn:aws:controltower:ap-northeast-3::control/ QVTIUVOTIXLQ</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Europe (Milan)            | Arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.EC2.PR.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-21</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ SZOFEUATQICK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------</td>
<td>-------------------</td>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td></td>
<td>east-2::control/ KUQMAGPMTIYY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ LPPLQ0DCGROV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ BFYFBNGDUPMY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ WTY1JXIQQ8WX</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ HVUXMMLIWFUT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ ESEU1NWAQPFU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ KMDVQBYAHYQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ ARDGJZTPMFD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ FKBVRKH0NP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ PTTXHPKISJL</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ DMNHGNNKZEBP</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ XNJAYSGVGVM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>west-3::control/ZZTQVTTCXGIR</td>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/LCMLSUBEDNHH</td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/YCYMJSOVKIMD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/XRRNGSVWUOX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/LCZICULVYVGTQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/AFMPKBWMAQVX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/DBCLUROPTPQJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/KHRCV5MBFTBG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/CNOBPUW3KIGF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/LSMWAQUJVBDF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/UCCVQSNPYVQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/GAOYRTVQGHB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>CT.EC2.PR.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-21</td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ CUDBAEGBAETL</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ BOOVPGTDMMV</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ JAUAVPLUXQF</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ SLQHJSMHMF</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ NKVJXXQBVISF</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ JHBKWDQDGXY</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ DVZJISBFMPXQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/ LPNSPPZGRSQC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:me-central-1::control/ BEKNCPBBAOLVJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/ VDOHFFWLBMLL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/ HJBKWDQDGXY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/ DVZJISBFMPXQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/OTTRCZABNRLV</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/UKGRYSQWORG</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/XOGWRUHGLMYT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/RWBQBCW65MYZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/PPEWYJ2UJTDL</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/WSPJASM5SHIMH</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/RHJKQFJVLKFW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/JPXSBS5EPSWSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/VSYRJAIQJXRE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/ALCTCNZFFPWY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/ZRLPKT2VMADM</td>
</tr>
</tbody>
</table>
| | | | • Asia Pacific (Osaka) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ECR.PR.1</td>
<td></td>
<td></td>
<td>northeast-3::control/EBGERJIZRCPES</td>
</tr>
</tbody>
</table>
|                    | NIST 800-53 Rev 5 CA-9(1) | Manage vulnerabilities | Europe (Milan)  
ar:aws:controltower:eu-south-1::control/LLULMIXRNSIP  
Africa (Cape Town)  
ar:aws:controltower:af-south-1::control/UBKVJIXAHCA  
Middle East (Bahrain)  
ar:aws:controltower:me-south-1::control/NJPSHTUOAZO  
Israel (Tel Aviv)  
ar:aws:controltower:il-central-1::control/KQYEYPKZGL  
Europe (Zurich)  
ar:aws:controltower:eu-central-2::control/MLYQXYDSM0NZ  
Europe (Spain)  
ar:aws:controltower:eu-south-2::control/FMACBWHXOSO  
Asia Pacific (Hyderabad)  
ar:aws:controltower:ap-south-2::control/QCWEESSTSTC  
Middle East (UAE)  
ar:aws:controltower:me-central-1::control/HZGIRZSHINZB  
Asia Pacific (Melbourne)  
ar:aws:controltower:ap-southeast-4::control/BFPRPYTVROPT  

CT.ECR.PR.1
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>• Protect configurations</td>
<td>east-1::control/QXNBGIPKJBEQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2(2)</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/WUAWJINPVBIG</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.1</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/IRQTOSLASFTT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/IMGIIIRQPZYL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/QBVJWKFJGIXU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/UKSAVCPEYXYG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/BLRQAANKFPRK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/JHXDESJFVFUH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/GEXAFJDYDCYJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/LAUYCZEXCNRP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/MVOMRLIOXVFN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/QMYDBMPB00HM</td>
</tr>
</tbody>
</table>
| | | | • Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeast-1::control/IRDNDNOLWMNN</td>
<td>• Europe (Paris) (\text{arn:aws:controltower:eu-west-3::control/CXODZHXMVCVEE})</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo) (\text{arn:aws:controltower:sa-east-1::control/UERWGVPURZSY})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California) (\text{arn:aws:controltower:us-west-1::control/THF1WZWCZGV})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong) (\text{arn:aws:controltower:ap-east-1::control/EHPCWFMOJUEW})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta) (\text{arn:aws:controltower:ap-southeast-3::control/YBFQ0OUUSWSP})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka) (\text{arn:aws:controltower:ap-northeast-3::control/QZAKDBVRQPU})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan) (\text{arn:aws:controltower:eu-south-1::control/KIPJ0MKOBOMA})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town) (\text{arn:aws:controltower:af-south-1::control/IOIURJXAO7TB})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain) (\text{arn:aws:controltower:me-south-1::control/DIPXXRQWFAQW})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv) (\text{arn:aws:controltower:il-central-1::control/DKZSCADUVKR})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich) (\text{arn:aws:controltower:eu-central-2::control/GOVJWUEVROCT})</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CT.ECR.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ECR.PR.2</td>
<td></td>
<td>Manage vulnerabilities</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 RA-5</td>
<td></td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11.2.3</td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.3.2</td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Europe (Frankfurt)      |                 |                   | • Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/ITYUPRDRHQ1X                                                                                                                                                      |
| Europe (Ireland)       |                 |                   | • Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/WYFFKCKEPLJD                                                                                                                                                     |
| Europe (London)        |                 |                   | • Europe (London)   
arn:aws:controltower:eu-west-2::control/IEEUEQBFCCBK                                                                                                                                                       |
| Europe (Stockholm)     |                 |                   | • Europe (Stockholm) 
arn:aws:controltower:eu-north-1::control/CTBPMNOJRRRG                                                                                                                                                   |
| Asia Pacific (Mumbai)  |                 |                   | • Asia Pacific      
(Mumbai)  
arn:aws:controltower:ap-south-1::control/WBSBAJCHOSLR                                                                                                                                                     |
| Asia Pacific (Seoul)   |                 |                   | • Asia Pacific      
(Seoul)   
arn:aws:controltower:ap-northeast-2::control/ELSOUNERQMFA                                                                                                                                                   |
| Asia Pacific (Tokyo)   |                 |                   | • Asia Pacific      
(Tokyo)   
arn:aws:controltower:ap-northeast-1::control/BBNLHIKHVSDF                                                                                                                                                  |
| Europe (Paris)         |                 |                   | • Europe (Paris)    
arn:aws:controltower:eu-west-3::control/YBYDPISAFGVJ                                                                                                                                                         |
| South America (São Paulo) |                |                   | • South America     
(São Paulo) 
arn:aws:controltower:sa-east-1::control/OUGZGHPIBMEQ                                                                                                                                                     |
| US West (N. California) |                 |                   | • US West (N. California)  
arn:aws:controltower:us-west-1::control/SXENUGPYUKAD                                                                                                                                                           |
| Asia Pacific (Hong Kong) |                |                   | • Asia Pacific      
(Hong Kong)  
arn:aws:controltower:ap-east-1::control/PIRUMJAENJDF                                                                                                                                                     |
| Asia Pacific (Jakarta) |                 |                   | • Asia Pacific      
(Jakarta)  
arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| southeast-3::control/ NQTDLNZ60GBG | • Asia Pacific (Osaka)  
ar:n::aws:controltower:ap-northeast-3::control/ DSZQH0HPILVM | | |
| • Europe (Milan)  
ar:n::aws:controltower:eusouth-1::control/ IOCROUVHSSPK | | | |
| • Africa (Cape Town)  
ar:n::aws:controltower:af-south-1::control/ ZCFPKHNTVBL | | | |
| • Middle East (Bahrain)  
ar:n::aws:controltower:mesouth-1::control/ MDCLENDMMJGN | | | |
| • Israel (Tel Aviv)  
ar:n::aws:controltower:il-central-1::control/ OCSHRWCPBIGG | | | |
| • Europe (Zurich)  
ar:n::aws:controltower:eucentral-2::control/ BWSAXHIZEZDD | | | |
| • Europe (Spain)  
ar:n::aws:controltower:eusouth-2::control/ EAUQCAJHCRLS | | | |
| • Asia Pacific (Hyderabad)  
ar:n::aws:controltower:apsouth-2::control/ JUTYWVTNRCMS | | | |
| • Middle East (UAE)  
ar:n::aws:controltower:me-central-1::control/ PFYOBUVMABUR | | | |
| • Asia Pacific (Melbourne)  
ar:n::aws:controltower:ap-southeast-4::control/ QZESSATNRQBL | | | |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ECR.PR.3</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Protect configurations</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/RKDZMSUYQTH0</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ARYALUPZXERR</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-8(1)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/FQQEMCXBLGKN</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/FKSEYKP3JDJT</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.4</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/NEMIRSQYEPIE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/XZFSMLDDYHOK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eucentral-1::control/CZVDXRYERZBE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:europe-west-1::control/EVJNATWHXBW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:europe-west-2::control/ZEZBSLWHLNZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:europe-north-1::control/DQX05G0LEFTV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/QAMXL0USWQVO</td>
</tr>
</tbody>
</table>

1780
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ECNPMKMZLJXF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/KIXAFVFYWVOM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/FNBIPNPRWVQX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/FVDLDRMSGHME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/LQFQWXGRRJXG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/CPTYXYMLLOZE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/QRCFKIBEUPDE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/UQFGHQUAIBAI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/KIFLKMMDVPH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/FXWIRNEDJUV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/GRIJKKPPVEHJG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CT.ECS.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ECS.PR.1</td>
<td>NIST 800-53 Rev 5 AC-2(1)</td>
<td>Manage vulnerabilities</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/MHZENEKENVRDM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/MYSVOZBSJPI</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/PRTAZZJJIMQM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:can-central-1::control/GBPVRLBOQCKZ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-5</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-4::control/QGQ0OLGEUFRO</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-3::control/QGQ0OLGEUFRO</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/EEHWJWMDSOSE</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>Middle East (UAE) arn:aws:controltower:me-central-1::control/NPSTOIEYDVPV</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td>Asia Pacific (Zurich) arn:aws:controltower:eu-central-2::control/YAFJVSLTGFW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Spain) arn:aws:controltower:eu-south-2::control/UHSKPFCFWNHJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Spain) arn:aws:controltower:eu-central-2::control/YAFJVSLTGFW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Zurich) arn:aws:controltower:eu-central-2::control/YAFJVSLTGFW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Spain) arn:aws:controltower:eu-south-2::control/UHSKPFCFWNHJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/EEHWJWMDSOSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Middle East (UAE) arn:aws:controltower:me-central-1::control/NPSTOIEYDVPV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/QGQ0OLGEUFRO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-3::control/QGQ0OLGEUFRO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:can-central-1::control/GBPVRLBOQCKZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-3::control/QGQ0OLGEUFRO</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(2)</td>
<td></td>
<td></td>
<td>southeast-2::control/AELENOQZMXVF</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.7</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ARPWJUYLVOEF</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 6.2</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/MJYZZTPTBHOK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/FJIW5MFTNUJP</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/PYRRLZWZEGC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/JQR1G3X0XYPN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/KPQKCVFHKYM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/AVQAVL5BSYMU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/JJTKL5LWWGCU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.4</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/TFUJIAEBMCMO</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.5</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/LBRTCEKHUMKQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.6</td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/SYYXAN5PGYM0</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective, by Region</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------</td>
<td>-------------------------------</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/MZVXPGLPQWM</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/ORAWXHCEJW</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/RXWVKVANCUNSA</td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/JRISOKFFILDM</td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/NDMGEZJEXYO</td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/EBKWKZSXPRTF</td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/FYQMDAKQISNQ</td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/MUCKKRINLINT</td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/GMOHYFMJQWC</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/JUYGSGOUWJFL</td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/QJGKAZAATQHG</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Melbourne)</td>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.ECS.PR.10</td>
<td></td>
<td></td>
<td>southeast-4::control/OQTRAQXRMKF</td>
</tr>
</tbody>
</table>

### CT.ECS.PR.10

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ECS.PR.10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Protect configurations</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ S1JXAEJNCZRU</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2</td>
<td>Enforce least privilege</td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ KRBXESUGGRWX</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.7</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ TZNJBPCHXKPN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ QFQHDLEJLUVA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ BARUWERYCVPZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ JYWFVNVCIRVR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ NIQDIWOHUBIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ QTCQLTICJQZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ BCRKZG0D0L</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ XWUPVDSFNTVZ</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/TOAJYCRKCFXD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/OHEAQZBMADFE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/SFTBSYXOGMZG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ZBUMWEYBEHMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/XJRVRGJIFJMX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/NNYUGCATXOCK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/BDFWFMGKGTLC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/AJNXXRBUKMIW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/NTURYGMJMDN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/NUGUPHOMSEQV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/LUPHAYKCDNOT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.ECS.PR.11</td>
<td></td>
<td>• Enforce least privilege</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/FKKIKNEMQIMG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/YMJJSF0HTFV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/FNUUGNHLPMKT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-</td>
</tr>
</tbody>
</table>

• NIST 800-53 Rev 5 AC-2(1)
• NIST 800-53 Rev 5 AC-3
• NIST 800-53 Rev 5 AC-3(15)
• NIST 800-53 Rev 5 AC-3(7)
• NIST 800-53 Rev 5 AC-5
• NIST 800-53 Rev 5 AC-6
• PCI DSS version 3.2.1 7.1.1
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td></td>
<td>central-1::control/EINHSETMUTZS</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/AIYVMwDWTLCW</td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/EFCZNPV0MKNP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-southeast-3::control/WANJBLRHTDXZ</td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-northeast-3::control/QHUEUPKSMVAC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-northeast-3::control/QHUEUPKSMVAC</td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/VHG5wXFFXGEA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:af-south-1::control/VHG5wXFFXGEA</td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/HTCFOMOTT5OB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:me-south-1::control/HTCFOMOTT5OB</td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/DISVJHLQHY00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:il-central-1::control/DISVJHLQHY00</td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/TXZQcWTXJKSR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-central-2::control/TXZQcWTXJKSR</td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/YKEDJRVRZKEH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-2::control/YKEDJRVRZKEH</td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/UZYAASORZSTK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-south-2::control/UZYAASORZSTK</td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/MPHCEJEWQEZG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------</td>
<td>----------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.ECS.PR.12</td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Use strong authentication</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/AMSZNQJFMYFK</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/PDBTCIAMJJFJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/DFEQFDPQBGA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/UICVAWRLEUIA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/QANDDOVB0KJX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/EDPKYC GGMW5S</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/JWXCYUI PKH HA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/EMXLYJRHKCXQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/ ZIQTSJXBFAEB</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>Arn:aws:controltower:eu-north-1::control/XBKALBVBFBMU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>Arn:aws:controltower:ap-south-1::control/PHUSQUZKDFW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>Arn:aws:controltower:ap-northeast-2::control/JTQTBQTJXRTY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>Arn:aws:controltower:ap-northeast-1::control/MGGRRKOTQYGT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>Arn:aws:controltower:eu-west-3::control/FRGKEKWRNAFI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>Arn:aws:controltower:sa-east-1::control/ANOIHODOAZVF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>Arn:aws:controltower:us-west-1::control/YDLIOLKELUTQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>Arn:aws:controltower:ap-east-1::control/XAKUNACPPCIH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>Arn:aws:controltower:ap-southeast-3::control/FBN1UCYMFSFGP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>Arn:aws:controltower:ap-northeast-3::control/CUTNMQPJIESP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>Arn:aws:controltower:eu-south-1::control/LGDXTFQGGNDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>Arn:aws:controltower:af-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CT.ECS.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ECS.PR.2</td>
<td>NIST 800-53 Rev 5 AU-6(3)</td>
<td>• Establish logging and monitoring</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/BYFWXRYSYXBV</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/KERIBBICUEGG</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Middle East (Bahrain) arn:aws:controltower:me-south-1::control/CDBKAMWXZRHR
- Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/JMISFEYTYRYK
- Europe (Zurich) arn:aws:controltower:eucentral-2::control/RCKSOE0J3J3XQ
- Europe (Spain) arn:aws:controltower:eusouth-2::control/HCDN453UJYVIB
- Asia Pacific (Hyderabad) arn:aws:controltower:apsouth-2::control/DIMYMTIRUMAR
- Middle East (UAE) arn:aws:controltower:me-central-1::control/WIKFYNHQWXXI
- Asia Pacific (Melbourne) arn:aws:controltower:apsoutheast-4::control/JTJZEPHNXQVH
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
<td>west-2::control/LBPTWMIHFKYF</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/SHIXZVMGGYMM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/WXYORQOYZPPG</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/FDKSTHVUWWYA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/WRZGKFSXXORW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/HFXLZYQMGFRS</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.7</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/FOHVPKZVRHPF</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ZUDLVFPNDOIO</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/DOAWPTYPCTCV</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/LQMMPHYSFREQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/IUNRXFCQGPRI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/OIUCPBXNKYYM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>ZEHKMMLYIFGJ</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td>SGZTSNHFPGQDX</td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>LXJSSCOAPFZV</td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>WMYOLPNABXBV</td>
</tr>
<tr>
<td>(Jakarta)</td>
<td></td>
<td></td>
<td>Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>USNBRAEITOF</td>
</tr>
<tr>
<td>(Osaka)</td>
<td></td>
<td></td>
<td>Europe (Milan)</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>DQFKEHKZLQDW</td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td></td>
<td>YHHQMSOSLWSQ</td>
</tr>
<tr>
<td>(Bahrain)</td>
<td></td>
<td></td>
<td>Africa (Cape Town)</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>GBBHLSBWCWQs</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>KKQJCJDMOLM</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>MNKWNXTMDNM</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>UFMSTKPVLYKF</td>
</tr>
<tr>
<td>(Hyderabad)</td>
<td></td>
<td></td>
<td>Asia Pacific (Hyderabad)</td>
</tr>
</tbody>
</table>
## CT.ECS.PR.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ECS.PR.3</td>
<td>• NIST 800-53 Rev 5 AC-2(1) &lt;br&gt;• NIST 800-53 Rev 5 AC-3 &lt;br&gt;• NIST 800-53 Rev 5 AC-3(15) &lt;br&gt;• NIST 800-53 Rev 5 AC-5 &lt;br&gt;• NIST 800-53 Rev 5 AC-6 &lt;br&gt;• PCI DSS version 3.2.1 7.1.1</td>
<td>• Enforce least privilege &lt;br&gt;• Manage vulnerabilities</td>
<td>• US East (N. Virginia) &lt;br&gt;arn:aws:controltower:us-east-1::control/GFAVSWYHDTND &lt;br&gt;• US East (Ohio) &lt;br&gt;arn:aws:controltower:us-east-2::control/QHAHUINVTWCK &lt;br&gt;• US West (Oregon) &lt;br&gt;arn:aws:controltower:us-west-2::control/FCDPZLCZQNUW &lt;br&gt;• Canada (Central) &lt;br&gt;arn:aws:controltower:ca-central-1::control/WHBVGSRPVIND &lt;br&gt;• Asia Pacific (Sydney) &lt;br&gt;arn:aws:controltower:ap-southeast-2::control/LHNKHNNMIXX &lt;br&gt;• Asia Pacific (Singapore) &lt;br&gt;arn:aws:controltower:ap-southeast-1::control/UWMACLLLDWQA &lt;br&gt;• Europe (Frankfurt) &lt;br&gt;arn:aws:controltower:eu-central-1::control/ICGGLMZIPMAT &lt;br&gt;• Europe (Ireland) &lt;br&gt;arn:aws:controltower:eu-west-1::control/LQBIGKSRNNIXX</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/AROELFZBBOTF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/XUQCMRRSTMWQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/RQJOQNGJDIYW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/YSVBKSPQRORD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/BKVHTGKQKEYEU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/EFDXHWMRVIRK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/RQWURAFSPNKR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/QIGZSAMESKYV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/NOIWGZCKJWDWV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/HIVKSROFOHMI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/HIAVTIWWMCV</td>
</tr>
</tbody>
</table>
|                   |           |                   | • Europe (Milan) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ECS.PR.4</td>
<td>NIST 800-53 Rev 5 SC-7</td>
<td>Limit network access</td>
<td>• US East (N. Virginia) : arn:aws:controltower:us-east-1::control/ YAGJMMJCIOQDQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td>• US East (OH) : arn:aws:controltower:us-</td>
</tr>
</tbody>
</table>

**Control identifier**

**Framework**

**Control objective**

**Control API identifiers, by Region**

- **Africa (Cape Town)**: arn:aws:controltower:af-south-1::control/ REYGPHBQWSGI
- **Middle East (Bahrain)**: arn:aws:controltower:me-south-1::control/ PXEGBZWRQERK
- **Israel (Tel Aviv)**: arn:aws:controltower:il-central-1::control/ SYGIWSEDSEZZ
- **Europe (Zurich)**: arn:aws:controltower:eu-central-2::control/ AAWOAIJMXYLNY
- **Europe (Spain)**: arn:aws:controltower:eu-south-2::control/ PLGSGRFMDCMHH
- **Asia Pacific (Hyderabad)**: arn:aws:controltower:ap-south-2::control/ ACWEVNIMYUCL
- **Middle East (UAE)**: arn:aws:controltower:me-central-1::control/ DKIXMNQCFEDO
- **Asia Pacific (Melbourne)**: arn:aws:controltower:ap-southeast-4::control/ EYTOIVASPVEM
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td>east-2::control/ MGBLOFCADDWF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ NDQWPPNXIKJ0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ IARHIPWNLNY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ NRRNSUVGQA0K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ DQLFFJBMHJJB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ NAMUQZRYR3Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ KOMKX1QGQHVI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ OZAVAEBPZ XP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ ETHWPQFJNBHN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ ZAWNBERLIWEW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ IBTHPPTAYK</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ GODEVZ5HFXUC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective, by Region</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------------------</td>
<td></td>
</tr>
<tr>
<td>west-3::control/</td>
<td></td>
<td>NMYHLHTRRSMF</td>
<td></td>
</tr>
<tr>
<td>• South America</td>
<td></td>
<td>(São Paulo) arn:aws:controltower:sa-east-1::control/DNLNJTIGMMFI</td>
<td></td>
</tr>
<tr>
<td>South America</td>
<td></td>
<td>• US West (N. California)</td>
<td></td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td>California) arn:aws:controltower:us-west-1::control/FMYFLIBUAOQB</td>
<td></td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td>(Hong Kong) arn:aws:controltower:ap-east-1::control/RHQTXEGVONX</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific</td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td>(Jakarta) arn:aws:controltower:ap-southeast-3::control/RVWNBZEIAMPE</td>
<td></td>
</tr>
<tr>
<td>Osaka)</td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td>(Osaka) arn:aws:controltower:ap-northeast-3::control/VAAGGLZEFDKT</td>
<td></td>
</tr>
<tr>
<td>Osaka)</td>
<td></td>
<td>• Europe (Milan)</td>
<td></td>
</tr>
<tr>
<td>• Europe</td>
<td></td>
<td>(Milan) arn:aws:controltower:eu-south-1::control/NZWMBPGYEQI</td>
<td></td>
</tr>
<tr>
<td>Europe</td>
<td></td>
<td>• Africa (Cape Town)</td>
<td></td>
</tr>
<tr>
<td>• Africa</td>
<td></td>
<td>(Cape Town) arn:aws:controltower:af-south-1::control/DDEZLVOWZIVH</td>
<td></td>
</tr>
<tr>
<td>Africa</td>
<td></td>
<td>• Middle East (Bahrain)</td>
<td></td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td>(Bahrain) arn:aws:controltower:me-south-1::control/OZRRTWBBIQJ</td>
<td></td>
</tr>
<tr>
<td>Middle East</td>
<td></td>
<td>• Israel (Tel Aviv)</td>
<td></td>
</tr>
<tr>
<td>• Israel</td>
<td></td>
<td>(Tel Aviv) arn:aws:controltower:il-central-1::control/HZKZHDATTHMM</td>
<td></td>
</tr>
<tr>
<td>Israel</td>
<td></td>
<td>• Europe (Zurich)</td>
<td></td>
</tr>
<tr>
<td>• Europe</td>
<td></td>
<td>(Zurich) arn:aws:controltower:eu-central-2::control/CWHSHXCPXHAN</td>
<td></td>
</tr>
<tr>
<td>Europe</td>
<td></td>
<td>• Europe (Spain)</td>
<td></td>
</tr>
<tr>
<td>• Europe</td>
<td></td>
<td>(Spain) arn:aws:controltower:eu-south-2::control/ZVTRPSJBSKGG</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/YIGLAGAFXXJ0</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/LERIWIHIZLWV</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/MBYLGYRJVVDQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/00ZUIEFUETGL</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/YAHLRLBBTDI</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/YQOFKMHUBER</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 10.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 10.2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 10.2.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CT.ECS.PR.5

- **Control identifier**: CT.ECS.PR.5
- **Framework**
  - NIST 800-53 Rev 5 AC-4(26)
  - NIST 800-53 Rev 5 AU-10
  - NIST 800-53 Rev 5 AU-12
  - NIST 800-53 Rev 5 AU-2
  - NIST 800-53 Rev 5 AU-3
  - NIST 800-53 Rev 5 AU-6(3)
  - NIST 800-53 Rev 5 AU-6(4)
  - NIST 800-53 Rev 5 CA-7
  - NIST 800-53 Rev 5 SC-7(9)
  - NIST 800-53 Rev 5 SI-7(8)
  - PCI DSS version 3.2.1 10.1
  - PCI DSS version 3.2.1 10.2.1
  - PCI DSS version 3.2.1 10.2.2
  - PCI DSS version 3.2.1 10.2.3
- **Control objective**: Establish logging and monitoring
- **Control API identifiers, by Region**
  - **Asia Pacific (Hyderabad)** arn:aws:controltower:ap-south-2::control/ DZMBACOEVMXS
  - **Middle East (UAE)** arn:aws:controltower:me-central-1::control/ XWNOQNOXO9VF
  - **Asia Pacific (Melbourne)** arn:aws:controltower:ap-southeast-4::control/ YTNYDNTBJROR
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/DEAHTYWKLWPU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/DSNUNMWDLPJA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.6</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/YAUDJTJ3YGTZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.7</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/AEKSNNGMNVUYJ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/JLPDEEKGTNS</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/NWTUHESHFSA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/XCQQPQZSYJQL</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/NHSQHFUTGNCN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/SEKFIMJRALUDZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/AFXFIKNNJTCD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/NXRQVWELREZ</td>
</tr>
</tbody>
</table>
| | | | • Asia Pacific (Osaka) arn:aws:controltower:ap-
### CT.ECS.PR.6

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td></td>
<td>east-1::control/JTWGVRQVPNPA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US East (Ohio)</td>
<td>arn:aws:controltower:us-east-2::control/WXRVLDXHXYTSD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td>arn:aws:controltower:us-west-2::control/WPJYPYBZNNHQO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td>arn:aws:controltower:ca-central-1::control/IFZZ0ZKBYHYXF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/ALOSYSCDXMC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/PEWMZWHWAXNB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/XPAYGNHWXKPRQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/KVIHSHXVHFCX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/OPDRCLSYBLYK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/PWQAOGQMGWXY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/DNUFP3WDBSLZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/AMHLJICMIG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo)       | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| northea... | northeast-1::control/ | • Europe (Paris) | ZXDGEZSOPFCD
| | ZXDGEZSOPFCD | arn:aws:controltower:eu-west-3::control/ | SXGYEZKKJTY
| | | South America (São Paulo) | arn:aws:controltower:sa-east-1::control/ KFTLBREXIINM
| | | US West (N. California) | arn:aws:controltower:us-west-1::control/ APHTPEHOKCKF
| | | Asia Pacific (Hong Kong) | arn:aws:controltower:ap-east-1::control/ CBOMKCLTZDHU
| | | Asia Pacific (Jakarta) | arn:aws:controltower:ap-southeast-3::control/ KWSNCTWDCYIA
| | | Asia Pacific (Osaka) | arn:aws:controltower:ap-northeast-3::control/ DZPUWGAWHSXH
| | | Europe (Milan) | arn:aws:controltower:eu-south-1::control/ ONBILKCURYST
| | | Africa (Cape Town) | arn:aws:controltower:af-south-1::control/ HORNBMWCMJORW
| | | Middle East (Bahrain) | arn:aws:controltower:me-south-1::control/ GBCMAONMPVEZ
| | | Israel (Tel Aviv) | arn:aws:controltower:il-central-1::control/ JTYEXQVOZSBN
| | | Europe (Zurich) | arn:aws:controltower:eu-central-2::control/ VKEZRWNMAPCG
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.ECS.PR.7         |           | Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/KXUMGTKTEKZR
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/IPDXJYKWODXA
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/RUYZPTAVRFN
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/OWWEUIDVOUHP
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/NEZKAVUMHCLM
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/REWMTIUNPNGU |

| CT.ECS.PR.7         | NIST 800-53 Rev 5 CA-9(1) | Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/KXUMGTKTEKZR
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/IPDXJYKWODXA
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/RUYZPTAVRFN
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/OWWEUIDVOUHP
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/NEZKAVUMHCLM
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/REWMTIUNPNGU |
| CT.ECS.PR.7         | NIST 800-53 Rev 5 CM-2 | Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/KXUMGTKTEKZR
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/IPDXJYKWODXA
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/RUYZPTAVRFN
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/OWWEUIDVOUHP
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/NEZKAVUMHCLM
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/REWMTIUNPNGU |
| CT.ECS.PR.7         | PCI DSS version 3.2.1 2.2 | Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/KXUMGTKTEKZR
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/IPDXJYKWODXA
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/RUYZPTAVRFN
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/OWWEUIDVOUHP
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/NEZKAVUMHCLM
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/REWMTIUNPNGU |
| CT.ECS.PR.7         | NIST 800-53 Rev 5 CA-9(1) | Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/KXUMGTKTEKZR
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/IPDXJYKWODXA
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/RUYZPTAVRFN
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/OWWEUIDVOUHP
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/NEZKAVUMHCLM
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/REWMTIUNPNGU |
| CT.ECS.PR.7         | NIST 800-53 Rev 5 CM-2 | Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/KXUMGTKTEKZR
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/IPDXJYKWODXA
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/RUYZPTAVRFN
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/OWWEUIDVOUHP
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/NEZKAVUMHCLM
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/REWMTIUNPNGU |
| CT.ECS.PR.7         | PCI DSS version 3.2.1 2.2 | Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/KXUMGTKTEKZR
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/IPDXJYKWODXA
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/RUYZPTAVRFN
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/OWWEUIDVOUHP
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/NEZKAVUMHCLM
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/REWMTIUNPNGU |

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.ECS.PR.7         | NIST 800-53 Rev 5 CA-9(1) | Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/KXUMGTKTEKZR
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/IPDXJYKWODXA
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/RUYZPTAVRFN
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/OWWEUIDVOUHP
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/NEZKAVUMHCLM
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/REWMTIUNPNGU |
| CT.ECS.PR.7         | NIST 800-53 Rev 5 CM-2 | Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/KXUMGTKTEKZR
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/IPDXJYKWODXA
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/RUYZPTAVRFN
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/OWWEUIDVOUHP
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/NEZKAVUMHCLM
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/REWMTIUNPNGU |
| CT.ECS.PR.7         | PCI DSS version 3.2.1 2.2 | Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/KXUMGTKTEKZR
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/IPDXJYKWODXA
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/RUYZPTAVRFN
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/OWWEUIDVOUHP
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/NEZKAVUMHCLM
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/REWMTIUNPNGU |

Frame: CT.ECS.PR.7

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| Europe (Spain)     | Improve availability | • Europe (Spain)  
arn:aws:controltower:eu-south-2::control/JIZJUWRJAZLR |
| Asia Pacific (Hyderabad) | Improve availability | • Asia Pacific (Hyderabad)  
arn:aws:controltower:ap-south-2::control/RYOJTKATLZWB |
| Middle East (UAE)  | Improve availability | • Middle East (UAE)  
arn:aws:controltower:me-central-1::control/CLYTXBYEVHWWI |
| Asia Pacific (Melbourne) | Improve availability | • Asia Pacific (Melbourne)  
arn:aws:controltower:ap-southeast-4::control/UWJWVKORWUOI |
| US East (N. Virginia) | Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/KXUMGTKTEKZR |
| US East (Ohio)     | Improve availability | • US East (Ohio)  
arn:aws:controltower:us-east-2::control/IPDXJYKWODXA |
| US West (Oregon)   | Improve availability | • US West (Oregon)  
arn:aws:controltower:us-west-2::control/RUYZPTAVRFN |
| Canada (Central)   | Improve availability | • Canada (Central)  
arn:aws:controltower:ca-central-1::control/OWWEUIDVOUHP |
| Asia Pacific (Sydney) | Improve availability | • Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/NEZKAVUMHCLM |
| Asia Pacific (Singapore) | Improve availability | • Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/REWMTIUNPNGU |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/ CSEKQRBEBHI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/ SECTLWSVMKFB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ FSUOYQJVISCW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ SXCPWIDYPOPO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ HRFZRYYONQAJF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ ZHITNVQDDRNT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ BSSDTCTJXIDJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ IMYNHRNBGGDP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ RPTFPYNJVRRL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ UKIVIVVHEFYM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ DXBQECPIABVE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Asia Pacific (Jakarta) | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>southeeast-3::control/SZBHNXBFPDGI</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/QMHU05C6EHN</td>
</tr>
<tr>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/RSBGFQKAVQFR</td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/SERKY3MNJNYS</td>
</tr>
<tr>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/JUHBTVYAVHAF</td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/PVHIQKXYRMXE</td>
</tr>
<tr>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/DVTKBJDCSCIJ</td>
<td></td>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/R0OABJEGQ0BT</td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/FTVHCALGECX</td>
<td></td>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/ILMNQQLOTQKQF</td>
</tr>
<tr>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/PCIZQST6GKQ0Y</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CT.ECS.PR.8

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.ECS.PR.8        | • NIST 800-53 Rev 5 AC-2(1)  
                    • NIST 800-53 Rev 5 AC-3  
                    • NIST 800-53 Rev 5 AC-3(15)  
                    • NIST 800-53 Rev 5 AC-3(7)  
                    • NIST 800-53 Rev 5 AC-5  
                    • NIST 800-53 Rev 5 AC-6  
                    • PCI DSS version 3.2.1 7.1.1  
                    • PCI DSS version 3.2.1 7.1.2  
                    • PCI DSS version 3.2.1 7.2.1  
                    • PCI DSS version 3.2.1 7.2.2 | • Manage vulnerabilities |  
|                    |           |                  | • US East (N. Virginia)  
                    arn:aws:controltower:us-east-1::control/KKVIYLEZFGKT  
                    • US East (Ohio)  
                    arn:aws:controltower:us-east-2::control/RTPJRDZPWHEJ  
                    • US West (Oregon)  
                    arn:aws:controltower:us-west-2::control/GIGMOPPTKTBBKT  
                    • Canada (Central)  
                    arn:aws:controltower:ca-central-1::control/OPJANTDNJXUX  
                    • Asia Pacific (Sydney)  
                    arn:aws:controltower:ap-southeast-2::control/VAHLYXFO02HSE  
                    • Asia Pacific (Singapore)  
                    arn:aws:controltower:ap-southeast-1::control/JEGHYZBSMSQE  
                    • Europe (Frankfurt)  
                    arn:aws:controltower:eu-central-1::control/HEKFVPAUPMOV  
                    • Europe (Ireland)  
                    arn:aws:controltower:eu-west-1::control/DSIHJHPGNNQ  
                    • Europe (London)  
                    arn:aws:controltower:eu-west-2::control/LVFACZHPEFZ  
                    • Europe (Stockholm)  
                    arn:aws:controltower:eu-north-1::control/NLWRWUMEMNQ  
                    • Asia Pacific (Mumbai)  
                    arn:aws:controltower:ap-south-1::control/IGXZM0G0CQ8H |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/UCEBJDGYKZIW</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/KUHUYIOXNXL</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/YPNPABPVPNA</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/GCCOHHGINCEN</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/VYVWURVYUKKH</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/FRJMQPFLSYSC</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/CPUSPDMIROHK</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/NBTIIWMCBBVP</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/VQYCXZPSIXD</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/SMFMCNFPQLZ0</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/EBTYHGUHIJME</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-</td>
</tr>
</tbody>
</table>
## CT.ECS.PR.9

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ECS.PR.9</td>
<td>NIST 800-53 Rev 5</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>AC-21</td>
<td>• Enforce least privilege</td>
<td>arn:aws:controltower:us-east-1::control/</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td>ULWRYYUOYETK</td>
</tr>
<tr>
<td></td>
<td>AC-3</td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/</td>
</tr>
<tr>
<td></td>
<td>AC-3(7)</td>
<td></td>
<td>EFAKHMKKXWXMM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td>AC-4</td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td>OJMRKKE0NYNH</td>
</tr>
<tr>
<td></td>
<td>AC-4(21)</td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/</td>
</tr>
<tr>
<td></td>
<td>AC-6</td>
<td></td>
<td>QQBFQY1YOMAO</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td>SC-7</td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td>1810</td>
</tr>
<tr>
<td></td>
<td>SC-7(11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td>southeast-2::control/YXLNPXSEMTR</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/BZRONDFGDFUHD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/ZODTFHONMPAK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/UIJIYCPZPSNDW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/GSVMYFTSGKWB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/JHDQJBIOETHT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/BVARTMYMOPNP</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3.4</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/FKQVIWLKBLWC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3.6</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.4.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/UFCFASRBMGCT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.4.2</td>
<td></td>
<td></td>
<td>Europe (Paris)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.4.3</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/JDNTXDOQRRAGR</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.4.4</td>
<td></td>
<td></td>
<td>South America (São Paulo)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.4.5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/TIRWNJLSDRJSS</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.4.6</td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.4.7</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/BNESJVBSSJKKM</td>
</tr>
</tbody>
</table>

1811
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ CIRJDRQJIHRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ NOCAIXNKAIJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ KWFPPYEUQJKJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/ VMOPJVSAEBDK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/ KFPXJHOPUQJK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/ WJSSIZCTYILH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/ VCEDBHTMDGIK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/ PQYTNJTRMQG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/ CTEIMJAETBLH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/ ETUVKDVBNXWK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/ XAJBOXSRRCZS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Asia Pacific (Melbourne) | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.EKS.PR.1</td>
<td>NIST 800-53 Rev 5 AC-21</td>
<td>Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/WBFBCXMVAQM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/DYHEOMUQFX</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/GSLRTSNFSMMH</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/AWMWFFQFZKMX</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/NJSROBYVJRTO</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/SBDBCVUFDDXN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/NOIJJYYRQMGEU</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/MAKA5QNHXIS</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/OGZYDIHGSKFX</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/SHJWJBNMDRRG</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>southeaste-4:control/BNFDUDZYPCFD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td>• Asia Pacific (Mumbai) &lt;br&gt;arn:aws:controltower:ap-south-1::control/YHWWFD2UJRJRBQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Seoul) &lt;br&gt;arn:aws:controltower:ap-northeast-2::control/TGCCACSYPMRO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo) &lt;br&gt;arn:aws:controltower:ap-northeast-1::control/OZXLRJKGCMB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Paris) &lt;br&gt;arn:aws:controltower:eu-west-3::control/VWNLWLMKGRGC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo) &lt;br&gt;arn:aws:controltower:sa-east-1::control/ISUUFYPPTGNN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• US West (N. California) &lt;br&gt;arn:aws:controltower:us-west-1::control/ZZKJNURKGFHZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hong Kong) &lt;br&gt;arn:aws:controltower:ap-east-1::control/LRJDB0E0TGP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Jakarta) &lt;br&gt;arn:aws:controltower:ap-southeast-3::control/GTUICYBDWHLH0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Osaka) &lt;br&gt;arn:aws:controltower:ap-northeast-3::control/HSZAZLRVWDLR</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Milan) &lt;br&gt;arn:aws:controltower:eu-south-1::control/XJCBE6DKRKFOA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Africa (Cape Town) &lt;br&gt;arn:aws:controltower:af-south-1::control/GBJGALFTTBMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (Bahrain) &lt;br&gt;arn:aws:controltower:me-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CT.EKS.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.EKS.PR.2</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ ZYM01FKNKWNKL</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ TINBDMQQLKET</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| central-1::control/ NTRRBCOKHJHZ | • Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/KIPWXYPSTXZU |  |  |
| • Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/RRVFDZMDHUYX |  |  |  |
| • Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/XEMVRMKXHMTJ |  |  |  |
| • Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/PFWXKKGVZUKD |  |  |  |
| • Europe (London)  
arn:aws:controltower:eu-west-2::control/XKF1HAWINDZC |  |  |  |
| • Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/KMJXNNELEYEJC |  |  |  |
| • Asia Pacific (Mumbai)  
arn:aws:controltower:ap-south-1::control/KOPLFFVKEADB |  |  |  |
| • Asia Pacific (Seoul)  
arn:aws:controltower:ap-northeast-2::control/WYEYVIIZOLIL |  |  |  |
| • Asia Pacific (Tokyo)  
arn:aws:controltower:ap-northeast-1::control/BYEGHPQCSYN |  |  |  |
| • Europe (Paris)  
arn:aws:controltower:eu-west-3::control/YB0FKAFGRJZB |  |  |  |
| • South America (São Paulo)  
arn:aws:controltower:sa-east-1::control/00XFRUOXIKR |  |  |  |
<p>| • US West (N. California) |  |  |  |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>arn:aws:controltower:us-west-1::control/AQWSDBJGVPGE</td>
<td></td>
<td></td>
<td>- Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/WOYZKWUBDWG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/XCCMMBTOHCIV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/RHHGAMNJ2HYK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Europe (Milan) arn:aws:controltower:eu-south-1::control/DFGGQVUCOSMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Africa (Cape Town) arn:aws:controltower:af-south-1::control/PINKRROCTZKV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Middle East (Bahrain) arn:aws:controltower:me-south-1::control/EPVTQXKDNAJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/EXKWXMEDEQA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Europe (Zurich) arn:aws:controltower:eu-central-2::control/JVQULLJDBYJM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Europe (Spain) arn:aws:controltower:eu-south-2::control/GKQONVLDDFSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/DHNVB1PNLTTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Middle East (UAE) arn:aws:controltower:me-central-1::control/XDDWTJRHP0DP</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>CT.ELASTICACHE.PR.1</td>
<td>NIST 800-53 Rev 5 CP-10</td>
<td>Improve resiliency</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/KHOAPVGNWCL</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-6</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/SZITQJZTSCDZ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-9</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/WAGVKTPOVMVF</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/SIDZ0IYXVIKC</td>
</tr>
<tr>
<td></td>
<td>SI-12</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/PGZSVHJPXLRN</td>
</tr>
<tr>
<td></td>
<td>SI-13(5)</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/MMGPVLCCQDKC</td>
</tr>
<tr>
<td></td>
<td>SC-5(2)</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/NUTMZSMQDWKN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/IIYFHPBWXAWV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/APHCLUJKHADW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/KMRGWHJRKYI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/KHOAPVGNWCL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/SZITQJZTSCDZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/WAGVKTPOVMVF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/SIDZ0IYXVIKC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/PGZSVHJPXLRN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/MMGPVLCCQDKC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/NUTMZSMQDWKN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/IIYFHPBWXAWV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/APHCLUJKHADW</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eunortheast-1:control/OGAZAAWMPVYYP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:apsouth-1:control/RZRIXZMAVLI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:apnortheast-2:control/GCJWCUSXQIGJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:apnortheast-1:control/IKOCSATHALLQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:euwest-3:control/XEXHSVEAHLHK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:saeast-1:control/JNMEUSLGWULR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:uswest-1:control/IGPXRJTNACPW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:apeast-1:control/ZKJYXVYSXGIV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:apsoutheast-3:control/JBFIPVGUHGW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:apnortheast-3:control/LFDJWCRGQX CZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eusouth-1:control/EWNCZN FJJ EHH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:africa-central-1:control/AFCARTJGRQ</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# CT.ELASTICACHE.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| **CT.ELASTICACHE.PR.2** | • NIST 800-53 Rev 5 SI-2  
• NIST 800-53 Rev 5 SI-2(2)  
• NIST 800-53 Rev 5 SI-2(4)  
• NIST 800-53 Rev 5 SI-2(5) | • Manage vulnerabilities | • US East (N. Virginia)  
arin:aws:controltower:us-east-1::control/PRNLSSJSXSYHF  
• US East (Ohio)  
arin:aws:controltower:us-east-2::control/CEUMAASLVC9Z  
• US West (Oregon)  
arin:aws:controltower:us-west-1::control/K9H8NU6LQ987  
• Asia Pacific (Hyderabad)  
arin:aws:controltower:ap-south-2::control/TRJRZVZCRWLT  
• Middle East (UAE)  
arin:aws:controltower:me-central-1::control/JWHOBDXLVGHM  
• Asia Pacific (Melbourne)  
arin:aws:controltower:ap-southeast-4::control/FPZDNGAEMAN  
• Middle East (Bahrain)  
arin:aws:controltower:me-central-1::control/HQQRGAEOFJZL  
• Israel (Tel Aviv)  
arin:aws:controltower:il-central-1::control/OPJEYWQWCMZ  
• Europe (Zurich)  
arin:aws:controltower:eu-central-2::control/DRIXCEWAZKMA  
• Europe (Spain)  
arin:aws:controltower:eu-south-2::control/KKTMCYDDNQIE  
<p>|</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 6.2</td>
<td></td>
<td></td>
<td>west-2::control/XYPKYXTSZWZY</td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/GXKOJVEIZCKJ</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/TIWUDRZWGRHU</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/BNZBYWHDDJAS</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/OWQYYBNTGQLF</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/TJNEZHWNISPFE</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/LDBLMYQDWYBP</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/WUTMKZRYTHQY</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/LFUNGCAFTSLK</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/Q6OJFAEUBQYZ</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/KZEHHSKPGKCS</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/CVPBOKRKCETX</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>XNILTNRZOMAS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/CFBKHWVNORLM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/DHPFFGWJJJU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/QRBXRUFFBUZK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/JAYMTBJNCZCX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/KNYWNNWJJJU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/HBFYZGBECJMI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/KKI6LJ5UJMB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/EYDZCWLSCHUC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/TQIZCRCYALKX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/PNHPEPRLPUN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/FSMTBMP0WFCC</td>
</tr>
</tbody>
</table>
### CT.ELASTICACHE.PR.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ELASTICACHE.PR.3</td>
<td>NIST 800-53 Rev 5 CP-10</td>
<td>Improve resiliency</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ MWVHFZSSZTIM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ PWIBHPGBWXYR</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ AGVBFPDHFSFH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ BPLCSUSNIYE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ EGNYZBJWMLRF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ YYDVCJSVRUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ GGCXYMJIRASV</td>
</tr>
</tbody>
</table>

• Middle East (UAE) arn:aws:controltower:me-central-1::control/ ZPYVCNCKBSSHF
• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/ YUWPABBCXFKEK
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/IPSVVUAWSJJKK</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/KITWCDGNIGAB</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/IYGWAKKVKOA0</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/FGFSUSQPLYJZ</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/MZFHEBYRUQX</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/MQKWJQJDJUBN</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/HWQTAFCAIEVD</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/QZJGKNPODVTR</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/BVLUZNVRSNMN</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/BGSOAUDWHLHT</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/GMSVMLABMCNO</td>
</tr>
</tbody>
</table>
| Europe (Milan)     |           |                  | arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ELASTICACHE.PR.4</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Encrypt data at rest</td>
<td><strong>US East (N. Virginia)</strong>&lt;br&gt;arn:aws:controltower:us-east-1::control/ LUPKJZXBMHBK</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td><strong>US East (Ohio)</strong>&lt;br&gt;arn:aws:controltower:us-east-2::control/ EZLWWUXADVVZ</td>
</tr>
</tbody>
</table>

**CTX.Elasticache.PR.4**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ELASTICACHE.PR.4</td>
<td>• Africa (Cape Town)&lt;br&gt;arn:aws:controltower:af-south-1::control/ BVTITOUCJXHE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Africa (Cape Town)&lt;br&gt;arn:aws:controltower:af-south-1::control/ BVTITOUCJXHE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (Bahrain)&lt;br&gt;arn:aws:controltower:me-south-1::control/ RXVIDRGOEWSV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (Bahrain)&lt;br&gt;arn:aws:controltower:me-south-1::control/ RXVIDRGOEWSV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Israel (Tel Aviv)&lt;br&gt;arn:aws:controltower:il-central-1::control/ SHIMSNUKBOEK</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Israel (Tel Aviv)&lt;br&gt;arn:aws:controltower:il-central-1::control/ SHIMSNUKBOEK</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Zurich)&lt;br&gt;arn:aws:controltower:eu-central-2::control/ MXKAVOHNDTFJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Zurich)&lt;br&gt;arn:aws:controltower:eu-central-2::control/ MXKAVOHNDTFJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Spain)&lt;br&gt;arn:aws:controltower:eu-south-2::control/ DHONPPGSALEU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Spain)&lt;br&gt;arn:aws:controltower:eu-south-2::control/ DHONPPGSALEU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hyderabad)&lt;br&gt;arn:aws:controltower:ap-south-2::control/ JZIUWCPVXPRG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hyderabad)&lt;br&gt;arn:aws:controltower:ap-south-2::control/ JZIUWCPVXPRG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (UAE)&lt;br&gt;arn:aws:controltower:me-central-1::control/ ETLWWUXMLUX</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (UAE)&lt;br&gt;arn:aws:controltower:me-central-1::control/ ETLWWUXMLUX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td></td>
<td>east-2::control/URUJVBXOUUMZ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/BOKFJHSIIKG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/RWFPPWLVDFXEU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AYHVEAYSGMHD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/MMZEEFKNKSMU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/CERICEOQBAKR</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/QNKPFQUCXQPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/GBGWHHGNPRTQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ZRZRJIUBBRWC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/EUMQYRAPDFTL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/FBOBFITTNYNLT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ATDPLKOFVLVCR</td>
</tr>
</tbody>
</table>
|                    |          |                   | Europe (Paris) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>west-3::control/ABKNGZGLILSP</td>
<td>• South America (São Paulo) arn:aws:controltower:saeast-1::control/IGGUQCFBP00X</td>
<td>• US West (N. California) arn:aws:controltower:uswest-1::control/FJOMQPXJSNJQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/AQMSLTABZIJN</td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/KLQYRUHZYFAN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/LFQASCUYICHI</td>
<td>• Europe (Milan) arn:aws:controltower:eusouth-1::control/GLCJXMIVAWFG</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/RGXKTK0YEKBI</td>
<td>• Middle East (Bahrain) arn:aws:controltower:mesouth-1::control/IBSRWUDBKB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:ilcentral-1::control/SQMPAJNFRKMK</td>
<td>• Europe (Zurich) arn:aws:controltower:eucentral-2::control/MTYVBATSPZDM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eusouth-2::control/SBCVIFZBXREP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
CT.ELASTICACHE.PR.5 | NIST 800-53 Rev 5 AC-17(2) | Encrypt data in transit | US East (N. Virginia) arn:aws:controltower:us-east-1::control/ SJSYMZXGBOAX
| NIST 800-53 Rev 5 AC-4 | | | US East (Ohio) arn:aws:controltower:us-east-2::control/ VMDJUBBWADIG
| NIST 800-53 Rev 5 IA-5(1) | | | US West (Oregon) arn:aws:controltower:us-west-2::control/ JEJMEQFMTEBU
| NIST 800-53 Rev 5 SC-12(3) | | | Canada (Central) arn:aws:controltower:ca-central-1::control/ DJMUOJNRFWVF
| NIST 800-53 Rev 5 SC-13 | | | Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ ZHSLHILXVOCO
| NIST 800-53 Rev 5 SC-23 | | | Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ DCCIQ30HRVIE
| NIST 800-53 Rev 5 SC-23(3) | | | Europe (Frankfurt) arn:aws:controltower:eucentral-1::control/ LICKFADISXMU
<p>| NIST 800-53 Rev 5 SC-7(4) | | |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/JFLVEMYTDYRU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/DZXWKHDHYDBNP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/MNTKSMSTRMMF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/HMBLNJNCTDHZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/PIUNMTKIFFJL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/XAEPUCUPNGCNO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/ATOCTJSQAZEO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/XJOBGXRAPGMF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/JVJBKO0YKGQM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/RTTOLXATLPIII</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/IFMXFKFLEXXP</td>
</tr>
</tbody>
</table>
|                    |           |                   | • Asia Pacific (Osaka) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ELASTICACHE.PR.6</td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-4-1::control/OGTAAYEZIPUMA</td>
</tr>
</tbody>
</table>

- **Control identifier**
  - CT.ELASTICACHE.PR.6

- **Framework**
  - NIST 800-53 Rev 5 AC-4

- **Control objective**
  - Limit network access

- **Control API identifiers, by Region**
  - US East (N. Virginia) arn:aws:controltower:us-4-1::control/OGTAAYEZIPUMA

- **Region Examples**
  - Europe (Milan) arn:aws:controltower:eu-south-1::control/FGZTNSLDWXCI
  - Africa (Cape Town) arn:aws:controltower:af-south-1::control/RCTLXAMLCHXC
  - Middle East (Bahrain) arn:aws:controltower:me-south-1::control/UGTZTWBDXPC
  - Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/HNFDCNZRXUCD
  - Europe (Zurich) arn:aws:controltower:eu-central-2::control/IYCXHIMVNDRA
  - Europe (Spain) arn:aws:controltower:eu-south-2::control/XKMBREUXJFVN
  - Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/JKIQQDHBWJG
  - Middle East (UAE) arn:aws:controltower:me-central-1::control/ECVLLKGJ3GKE0
  - Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/OGTAAYEZIPUMA
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td></td>
<td>east-1::control/ FQLTUDZMHLVZ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ QOMUCVTCNKUO</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ NNZFDSJPKDBD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ VATZKDSWZ5OA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ WBJIXMOKJKNB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ INJGAMRVOAJQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(5)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ BTLRRCFRMVAQP</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ HOAEBNBZXSMO</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ QCUPCDGPHTUR</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ MCWTNZUDDXPM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ LJWYGOMFTCTB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ RDYIPVICGWEI</td>
</tr>
</tbody>
</table>
| • PCI DSS version 3.2.1 1.3.6 | | | • Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeeast-1:control/AVPOWAHSUBEB</td>
<td>Northeast-1</td>
<td>Control Objective</td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/CHXTKAYAXVV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/XLKNEMCRIYIIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/RNFHWESIBLOO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/DZEWCFADZWCP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/OMPFCRWUDYXW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/VFZVDKOUURFD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/BTIKMDWEMMBX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/FGNJBRONGTOG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/EMJLKUDJLHIJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/SYOAJQOBJNXH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/MEJYWIYAMQME</td>
</tr>
</tbody>
</table>
### CT.ELASTICACHE.PR.7

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.ELASTICACHE.PR.7 | - NIST 800-53 Rev 5 AC-2(1)  
- NIST 800-53 Rev 5 AC-3  
- NIST 800-53 Rev 5 AC-3(15)  
- NIST 800-53 Rev 5 AC-3(7)  
- NIST 800-53 Rev 5 AC-6  
- PCI DSS version 3.2.1 7.1.1  
- PCI DSS version 3.2.1 7.2.1  
- PCI DSS version 3.2.1 7.2.2 | - Enforce least privilege | - US East (N. Virginia)  
ar:aws:controltower:us-east-1::control/  
GHMQDSHZXXZU  
- US East (Ohio)  
ar:aws:controltower:us-east-2::control/  
ZHDBZRDJKKGN  
- US West (Oregon)  
ar:aws:controltower:us-west-2::control/  
JESD2QANVQZE  
- Canada (Central)  
ar:aws:controltower:ca-central-1::control/  
KBOVLCAXAJEC  
- Asia Pacific (Sydney)  
ar:aws:controltower:ap-southeast-2::control/  
HBXNCBGUQCBZ  
- Asia Pacific (Singapore)  
ar:aws:controltower:ap-southeast-1::control/  
IWBQXQYVWQGA |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/ FXUKOTDG0INU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/ DLTKNKGPIG6CN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ FKXDPENTGVWA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ RXEUZUTYSHW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ EAWBFGLUSNPY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ NUECFXEMFWTE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ LZOPZBLREDVZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ ONHSUTRTTLUN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ MKBEEQVXZQOY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ VTCDEGPDTFNK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ QHAHUQDYIRI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Asia Pacific (Jakarta) | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>southeast-3::control/IQDCLULOMHIN</td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/MJULTZFOIKMK</td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/MKLXFKZCODXZLQ</td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/WLPSXAAIVLNT</td>
</tr>
<tr>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/ZZUFSTXGONB</td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/NWVNCXUTZDKC</td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/SPTKJOPDXWG</td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/YELOMBVITOQ</td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/EPMCDGUGOKVQ</td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/JKVIXAUOMYKL</td>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/QMVNDDBFTCLC</td>
<td></td>
</tr>
</tbody>
</table>
### Control identifier

CT.ELASTICBEANSTALK.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ELASTICBEANSTALK.PR.1</td>
<td>NIST 800-53 Rev 5 CA-7</td>
<td>Improve resiliency</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/TMNYYCDENFCPB</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/FHW0IKXTZJBA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/HTYYYSZCPWOZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/CIEI0QVQORV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/YHIWFCYQDARH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/EARSTRSLZBZC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/HWOODCNIHCJI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/HBFJSXSUIMW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/DIFPFSQQPDGJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/KIOXCGYZAMNP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/HDYHDDNUEOBS</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/BVOHDZJHEJIW</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/UPSVYOWFPEJI</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/WRZBLOAGJNMN</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/CCZWODZDVSQP</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/DCSCUPBYBBU</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/XOARMTAVRPGI</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/CFMVUMABXQQU</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/ZITTOMDIEKYZ</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/EXRORZBLXWS</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/DFPWBGPWPAWXB</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/CXLWTIKPGOU</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.ELASTICBEANSTALK.PR.2</td>
<td>NIST 800-53 Rev 5 SI-2</td>
<td>Manage vulnerabilities</td>
<td>• US East (N. Virginia) &lt;br&gt;arn:aws:controltower:us-east-1::control/CGCIVUYTOZICW</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-2(2)</td>
<td></td>
<td>• US East (Ohio)  &lt;br&gt;arn:aws:controltower:us-east-2::control/JPTNJBASRLLS</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-2(4)</td>
<td></td>
<td>• US West (Oregon)  &lt;br&gt;arn:aws:controltower:us-west-2::control/GFAFVGWBGKHH</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-2(5)</td>
<td></td>
<td>• Canada (Central)  &lt;br&gt;arn:aws:controltower:central-1::control/YYHZHDDXKKIA</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 6.2</td>
<td></td>
<td>• Asia Pacific (Sydney)  &lt;br&gt;arn:aws:controltower:ap-southeast-4::control/LCXDOOJWVVA</td>
</tr>
<tr>
<td>1838</td>
<td>CT.ELASTICBEANSTALK.PR.2</td>
<td>NIST 800-53 Rev 5 SI-2</td>
<td>• Europe (Zurich)  &lt;br&gt;arn:aws:controltower:eu-central-2::control/OQCJAPPBI1OM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-2(2)</td>
<td></td>
<td>• Europe (Spain)  &lt;br&gt;arn:aws:controltower:eu-south-2::control/JFCEIPDXBFEQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-2(4)</td>
<td></td>
<td>• Asia Pacific (Hyderabad)  &lt;br&gt;arn:aws:controltower:ap-south-2::control/MRWDNKOEJVS</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-2(5)</td>
<td></td>
<td>• Middle East (UAE)  &lt;br&gt;arn:aws:controltower:me-central-1::control/JPIKQDY5VC</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>southeast-2::control/UOCTRMADKUPW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-southeast-1::control/YIGECEKFFPJW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-central-1::control/NRXBKFAZGJEW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-west-1::control/TXCEPMIAUZF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-west-2::control/MVTLDACJZJS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-north-1::control/CXIWQGADTQOW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-south-1::control/QYIYEHUBOJKB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-northeast-2::control/VYCTCSCJQLYE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-northeast-1::control/VPNEUQLIRQZW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-west-3::control/NJTALLLWVREW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:sa-east-1::control/AHAMDQSHXDPXQ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/WEUXECSSLASH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ AORFBCGNCTRNN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ UFUQHFPHRDMMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ DNUgefZWZLId</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/ JMINTMKZQZXS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/ FRRHEGTSOLMJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/ ZBRMIHMFFZOKU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/ WHOFJAIXCLYJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/ SAPBZQTPSCWW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/ FBQCCIQKUJYH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/ YQ WedDIDGFtJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/ OHOIYXHVYJOK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Melbourne)</td>
<td>arn:aws:controltower:ap-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Control identifier | Framework | Control objective | Control API identifiers, by Region
---|---|---|---
CT.ELASTICBEANSTALK.PR.3 | NIST 800-53 Rev 5 | Establish logging and monitoring | southeeast-4::control/UFLZAXAQSSXDA

#### CT.ELASTICBEANSTALK.PR.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ELASTICBEANSTALK.PR.3</td>
<td>NIST 800-53 Rev 5</td>
<td>Establish logging and monitoring</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/SEQKNDNAZBENY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/SNODXMTMKMM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/MZMHWWZVDYDM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/VBLRNFNOUJAIX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/XPBHEZZNQJS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/SNATAZAYGYTDS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/HTMOVEUJDAMG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/TMMCJHMKVLCL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/YVBNROYSIQY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/WWEPCKUIJHFA</td>
</tr>
</tbody>
</table>

- NIST 800-53 Rev 5
  - AC-4(26)
  - AU-10
  - AU-12
  - AU-2
  - AU-3
  - AU-6(3)
  - AU-6(4)
  - CA-7
  - SC-7(9)
  - SI-7(8)
- PCI DSS version 3.2.1
  - 10.1
  - 10.2.1
  - 10.2.2
  - 10.2.3
  - 10.2.4
  - 10.2.5
  - 10.2.7
  - 10.3.1
  - 10.3.2
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) &lt;br&gt;arn:aws:controltower:ap-south-1::control/CBTNCZSTVKEE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) &lt;br&gt;arn:aws:controltower:ap-northeast-2::control/DCJBGPHZTLJW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) &lt;br&gt;arn:aws:controltower:ap-northeast-1::control/VDTCPKAYZEDK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td>• Europe (Paris) &lt;br&gt;arn:aws:controltower:eu-west-3::control/GCSYGTPYPSLQ</td>
</tr>
<tr>
<td>• Asia Pacific (São Paulo) &lt;br&gt;arn:aws:controltower:sa-east-1::control/NUBHZEECDULQ</td>
<td></td>
<td></td>
<td>• South America (São Paulo) &lt;br&gt;arn:aws:controltower:sa-east-1::control/NUBHZEECDULQ</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong) &lt;br&gt;arn:aws:controltower:ap-east-1::control/BZZGAALQHMG</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) &lt;br&gt;arn:aws:controltower:ap-east-1::control/BZZGAALQHMG</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta) &lt;br&gt;arn:aws:controltower:ap-southeast-3::control/DCNARSSNDLVD</td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) &lt;br&gt;arn:aws:controltower:ap-southeast-3::control/DCNARSSNDLVD</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka) &lt;br&gt;arn:aws:controltower:ap-northeast-3::control/BIJTQZZCVCXN</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) &lt;br&gt;arn:aws:controltower:ap-northeast-3::control/BIJTQZZCVCXN</td>
</tr>
<tr>
<td>• Europe (Milan) &lt;br&gt;arn:aws:controltower:eu-south-1::control/WQQMLOAYAPZQ</td>
<td></td>
<td></td>
<td>• Europe (Milan) &lt;br&gt;arn:aws:controltower:eu-south-1::control/WQQMLOAYAPZQ</td>
</tr>
<tr>
<td>• Africa (Cape Town) &lt;br&gt;arn:aws:controltower:af-south-1::control/WJOMTU0CASVN</td>
<td></td>
<td></td>
<td>• Africa (Cape Town) &lt;br&gt;arn:aws:controltower:af-south-1::control/WJOMTU0CASVN</td>
</tr>
<tr>
<td>• Middle East (Bahrain) &lt;br&gt;arn:aws:controltower:me-</td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) &lt;br&gt;arn:aws:controltower:me-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.ELASTICFILESYSYSTEM.PR.1 | NIST 800-53 Rev 5 CA-9(1) | Encrypt data at rest | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/DDRMDRSTNIU  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/ GHUIRLEMFQQA  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/DFNRMHHNSCYN  
• Canada (Central)  
arn:aws:controltower:ca- 

| CT.ELASTICFILESYSYSTEM.PR.1 | • Israel (Tel Aviv)  
arn:aws:controltower:il-central-1::control/XFHOTKGPELEI  
• Europe (Zurich)  
arn:aws:controltower:eucentral-2::control/LSNMTBHSXBR  
• Europe (Spain)  
arn:aws:controltower:eusouth-2::control/RGYDIAKRWXXN  
• Asia Pacific (Hyderabad)  
arn:aws:controltower:ap-south-2::control/NSKZNAONQADX  
• Middle East (UAE)  
arn:aws:controltower:me-central-1::control/BQOPUWCEPYXY  
• Asia Pacific (Melbourne)  
arn:aws:controltower:ap-southeast-4::control/JRRSPQBKXBIQ |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td>central-1::control/JEMASGVKBRQF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/QECOJWZQHMMQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ERMQRACXWHLO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/CFDLJLDURJFV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/BHMRGDCOJVNB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/MQVWCUIDHKMT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/NZNHLALWUMET</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/LJZEHFLGLFSO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ASIKFXCZHAZV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/HARKDEIGPSTW</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/LWIUILOQALZA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/YRAICDCVUJ0I</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• US West (N. California)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/CPMXJILCQZPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/FEUCYWTLF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/AUBCXBXVTKUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/GBVIFTEEMNCI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/NHKFJAHAGYJS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/LBDEBBXUISHG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/XRJYNTRNPOQG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/ORYVBIYCBRXI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/YZVALCXXASS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/JYAVIEWOClBTI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/MCSQEHQTIYwZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/GEJNURXWPWfG</td>
</tr>
</tbody>
</table>

1845
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ELASTICFILESYSYSTEM.PR.2</td>
<td>NIST 800-53 Rev 5 CP-10</td>
<td>• Improve resiliency</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-6(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-12</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 3.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Asia Pacific (Melbourne)
arn:aws:controltower:ap-southeast-4::control/CKKIPGIOQZLP

- US East (N. Virginia)
arwn:aws:controltower:us-east-1::control/AQYLHRCHTYFH
- US East (Ohio)
arwn:aws:controltower:us-east-2::control/ZHNDMAPIBQBP
- US West (Oregon)
arwn:aws:controltower:us-west-2::control/VEZVVVFVGFDL
- Canada (Central)
arwn:aws:controltower:ca-central-1::control/SMQITIAHFKPG
- Asia Pacific (Sydney)
arwn:aws:controltower:ap-southeast-2::control/VXMYEULBUKRK
- Asia Pacific (Singapore)
arwn:aws:controltower:ap-southeast-1::control/JVOHLVYUHCBK
- Europe (Frankfurt)
arwn:aws:controltower:eu-central-1::control/TTLCIBPZTALN
- Europe (Ireland)
arwn:aws:controltower:eu-west-1::control/EPULSMBTUMFD
- Europe (London)
arwn:aws:controltower:eu-west-2::control/MGIFGKQENSJD
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/SSHEKLSKMCES</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/AKIXEAUEXTWQ</td>
</tr>
<tr>
<td>(Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/YNQIXFRNQWF</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/OYKEDLNZHUMF</td>
</tr>
<tr>
<td>(Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/BDDOBOTQFTPN</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/NUHELTPITCRU</td>
</tr>
<tr>
<td>South America</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/FPUGNMHETPYB</td>
</tr>
<tr>
<td>(São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/RNKDEXRBVOON</td>
</tr>
<tr>
<td>US West (N.</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/RNKDEXRBVOON</td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/HJBQVLVYNJRHH</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/NUHELTPITCRU</td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/BDDOBOTQFTPN</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/TLQPHMZPCKTX</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.ELASTICFILESYSYSTEM.PR.3 | NIST 800-53 Rev 5 AC-6(10) | • Enforce least privilege | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/JVOEMQJALNEH  
• US East (Ohio) arn:aws:controltower:us-east-2::control/BIBZKVASTJJI  
• US West (Oregon) arn:aws:controltower:us-west-1::control/1848
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>west-2::control/LEXKWFEOQKTO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td>arn:aws:controltower:ca-central-1::control/RSZNNHZMPHGQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/FINCEVUKQTHU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/CPRQIRLLJJSZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/TRQSLRYSYASX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/NVRAHJAKGOWS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/SBVGEBNLOBWL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ETAFBHEUSISM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/RVTHUVTEDVMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/FJUCAMQRKDYC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/SCENHYSBARYU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/VEDDJXMCKQCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td>east-1::control/XPLGISTXYPWZ</td>
<td></td>
<td><strong>US West (N. California)</strong> arn:aws:controltower:us-west-1::control/IOLOGLDEUGC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Hong Kong)</strong> arn:aws:controltower:ap-east-1::control/LHJTTYMLMRBC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Jakarta)</strong> arn:aws:controltower:ap-southeast-3::control/RPSQLCSMRWE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Osaka)</strong> arn:aws:controltower:ap-northeast-3::control/OQHNNMODBGIR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Europe (Milan)</strong> arn:aws:controltower:eu-south-1::control/MGJEGYGDZACD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Africa (Cape Town)</strong> arn:aws:controltower:af-south-1::control/QKHCUSWAMJGM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Middle East (Bahrain)</strong> arn:aws:controltower:me-south-1::control/DYITJOJMCRW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Israel (Tel Aviv)</strong> arn:aws:controltower:il-central-1::control/XORRIUNYWEZX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Europe (Zurich)</strong> arn:aws:controltower:eu-central-2::control/VHDSEWDXCTU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Europe (Spain)</strong> arn:aws:controltower:eu-south-2::control/VLWKJNCQRRHV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Hyderabad)</strong> arn:aws:controltower:ap-south-2::control/CWJPIEJBGENM</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.ELASTICFILESYSYSTEM.PR.4 | NIST 800-53 Rev 5 AC-6(2) | Enforce least privilege | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/SOCNUAYXONKJ  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/BWCBIWTTHMBA  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/WTQPSFJEQ8EE  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/AKFECGTIWOJZ  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/THCBRFJBEVKS  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/JBBLWRTYSJII  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/HILXUVLMFYBM  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/GNOPUCJTMOMM  
• Middle East (UAE)  
arn:aws:controltower:me-central-1::control/UCZRABGXCLJE  
• Asia Pacific (Melbourne)  
arn:aws:controltower:ap-southeast-4::control/UOGIEQYUBVWF |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ NLNHKPUQRYEU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ ULQQLKFQFXG0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ XKQWNWPEIRFE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ TRASMSLGPGVZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ YSDPCITHZXQI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ TOVQIXNDOXW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ HIFLGMRIOCCZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ VMFHHQZIPJT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ JKB0CKTFDMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ TMVUNWZICUTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ ERXWMJAEFID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Europe (Milan)     | arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.ELASTICLOADBALANCING.PR.1 | NIST 800-53 Rev 5 AC-17(2) | Encrypt data in transit | • US East (N. Virginia)  
  arn:aws:controltower:us-east-1::control/
  RZVUTRWTVUEO  
• US East (Ohio)  
  arn:aws:controltower:us-east-2::control/
  QFIMBHVVNTCRP  
 | NIST 800-53 Rev 5 AC-4 |                           | • Asia Pacific (Hyderabad)  
  arn:aws:controltower:ap-south-2::control/
  CVGTZDJTZLJI  
• Middle East (UAE)  
  arn:aws:controltower:me-central-1::control/
  QFIMBHVVNTCRP  
• Asia Pacific (Melbourne)  
  arn:aws:controltower:ap-southeast-4::control/
  CGNKJZRYHIZJ  
 |                                                                                      | • Europe (Zurich)  
  arn:aws:controltower:eucentral-2::control/
  XJJFNWYXGEYG  
• Europe (Spain)  
  arn:aws:controltower:eusouth-2::control/
  EKLTMHIIJMJYR  
 |                                                                                      | • Africa (Cape Town)  
  arn:aws:controltower:afsouth-1::control/
  BQKDRWSQPQXA  
 |                                                                                      | • Middle East (Bahrain)  
  arn:aws:controltower:mesouth-1::control/
  RBIBSEJGZCG  
 |                                                                                      | • Israel (Tel Aviv)  
  arn:aws:controltower:ilcentral-1::control/
  LLDNCKCAOXDK  
 |                                                                                      | • Asia Pacific (Hyderabad)  
  arn:aws:controltower:ap-south-2::control/
  CVGTZDJTZLJI  
• Middle East (UAE)  
  arn:aws:controltower:me-central-1::control/
  QFIMBHVVNTCRP  
• Asia Pacific (Melbourne)  
  arn:aws:controltower:ap-southeast-4::control/
  CGNKJZRYHIZJ  
 |                                                                                      | • US East (N. Virginia)  
  arn:aws:controltower:us-east-1::control/
  RZVUTRWTVUEO  
• US East (Ohio)  
  arn:aws:controltower:us-east-2::control/
  QFIMBHVVNTCRP  
• Asia Pacific (Melbourne)  
  arn:aws:controltower:ap-southeast-4::control/
  CGNKJZRYHIZJ  

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(26)</td>
<td></td>
<td></td>
<td>east-2::control/ XMIEWIVADWDD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ BLNOUTJZPLAD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ GYIRQYZQGX0J</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ JWBLMBPMKKDD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ NKNJBLFCPCGJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ TFIIXKKSMTKDE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ HGMLJLFKUCFW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ VXIGTPWOJKYZ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ UKHAMPPHXYFV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ KDCYLYSTBLDFE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-10</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ SOSAA0OKSDIQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ MAVNERRCDGUI</td>
</tr>
</tbody>
</table>
| • NIST 800-53 Rev 5 IA-5(1) | | | Europe (Paris) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td>west-3::control/ FBDACBTEASNT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ ODINAEBQZJM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ LEZQMKJXCKRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ CTXVLAEDJDLJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ AXMNAMRZUWHT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ MYHWFMEYVGPL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/ CKLWEMXBFSKT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/ WAKLGKTSPGMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/ KBWQZTALIDXV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/ CYGZKWHILPSU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/ OJZJAELVLQQT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/ OLSPLUFATRIH</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1855
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.ELASTICLOADBALANCING.PR.10 | NIST 800-53 Rev 5 AC-4(26) | • Establish logging and monitoring | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/  
HUFENJFLXLUB  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/  
LAZXTTJXNZWX  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/  
MUNVwQKVNFII  
• Canada (Central)  
ar:n:aws:controltower:ca-central-1::control/  
CWGCUTVHELNV  
• Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/  
BARRVADARLXA  
• Asia Pacific (Singapore)  
ar:n:aws:controltower:ap-southeast-1::control/  
STTSKRCGXRNO  
• Europe (Frankfurt)  
ar:n:aws:controltower:eu-central-1::control/  
JBPMUDNREFNU |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1:control/VMAQQDYRDSVL</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2:control/PRRDNZJAOLXC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1:control/BYRDSECDQQJP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1:control/TWJWYQMUZOHV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2:control/DIAQAVDUUBSL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1:control/PBOVNVABHNQW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3:control/LUVNYZMNOBRU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1:control/XBYLAEZTWIQM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1:control/VUWZZZGVHLAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1:control/QRJNDNXHwVFN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3:control/CFXSCVNJURIC</td>
</tr>
</tbody>
</table>
| | | | Asia Pacific (Osaka) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ELASTICLOADBALANCING.PR.11</td>
<td>NIST 800-53 Rev 5</td>
<td>CA-9(1)</td>
<td>Improve resiliency</td>
</tr>
</tbody>
</table>

- **US East (N. Virginia)**
  - Control identifier: CT.ELASTICLOADBALANCING.PR.11
  - Control API identifier: arn:aws:controltower:us-east-1::control/MTFUXDNBVIJO

- **Europe (Milan)**
  - Control identifier: CT.ELASTICLOADBALANCING.PR.11
  - Control API identifier: arn:aws:controltower:europe-central-2::control/YJRVUXKIUKOK

- **Africa (Cape Town)**
  - Control identifier: CT.ELASTICLOADBALANCING.PR.11
  - Control API identifier: arn:aws:controltower:africa-south-1::control/YXPFXYZWCGBW

- **Middle East (Bahrain)**
  - Control identifier: CT.ELASTICLOADBALANCING.PR.11
  - Control API identifier: arn:aws:controltower:middle-east-central-1::control/YPFRBCBHDRCT

- **Israel (Tel Aviv)**
  - Control identifier: CT.ELASTICLOADBALANCING.PR.11
  - Control API identifier: arn:aws:controltower:il-central-1::control/XSCNLFTCFZKX

- **Europe (Zurich)**
  - Control identifier: CT.ELASTICLOADBALANCING.PR.11
  - Control API identifier: arn:aws:controltower:europe-central-2::control/YJRVUXKIUKOK

- **Europe (Spain)**
  - Control identifier: CT.ELASTICLOADBALANCING.PR.11
  - Control API identifier: arn:aws:controltower:europe-south-2::control/ZYFULHCEJDUY

- **Asia Pacific (Hyderabad)**
  - Control identifier: CT.ELASTICLOADBALANCING.PR.11
  - Control API identifier: arn:aws:controltower:asia-pacific-south-2::control/EFCETYXCCSQX

- **Middle East (UAE)**
  - Control identifier: CT.ELASTICLOADBALANCING.PR.11
  - Control API identifier: arn:aws:controltower:middle-east-central-1::control/HXXFE688WYYT

- **Asia Pacific (Melbourne)**
  - Control identifier: CT.ELASTICLOADBALANCING.PR.11
  - Control API identifier: arn:aws:controltower:asia-pacific-southeast-4::control/ADHRMNATNDFH
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CM-2               | NIST 800-53 Rev 5 | • US East (Ohio) arn:aws:controltower:us-east-2::control/DOVONHQVOITW  
                   |           | • US West (Oregon) arn:aws:controltower:us-west-2::control/YEHZEMPWSWH  
                   |           | • Canada (Central) arn:aws:controltower:ca-central-1::control/VGCEFHSLOXNT  
                   |           | • Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/RFHZP3REXLSX  
                   |           | • Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/GXOFHFAWFDVD  
                   |           | • Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/JQXOGQBYFKFU  
                   |           | • Europe (Ireland) arn:aws:controltower:eu-west-1::control/OTFLWYTQQQC  
                   |           | • Europe (London) arn:aws:controltower:eu-west-2::control/LCLXRDMMQG  
                   |           | • Europe (Stockholm) arn:aws:controltower:eu-north-1::control/TWBYWMNBMPNY  
                   |           | • Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/AMQCTRGRKRTZ  
                   |           | • Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/HPMPEDWYSHQ  
                   |           | • Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeate-1::control/SMFWP3QCLITI</td>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:europe-west-3::control/TIEXXSPKJPBM</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:south-america-east-1::control/VXULPCMAXNL</td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ALIDOHBLHSNDW</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:asia-pacific-east-1::control/YBRAEMTMLQQO</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:asia-pacific-southeast-3::control/YBEQEHQUOIBL</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:asia-pacific-northeast-3::control/USVNDLDKAFX</td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:europe-south-1::control/XFJ30ZQSMMDV</td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:africa-south-1::control/GRUSJXRHM</td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:middle-east-south-1::control/AVEPSXSIVSVP</td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:israel-central-1::control/OPKDCJIKNJSZ</td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:europe-central-2::control/PPJGURFCAKXK</td>
<td></td>
</tr>
</tbody>
</table>
## CT.ELASTICLOADBALANCING.PR.12

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.ELASTICLOADBALANCING.PR.12 | NIST 800-53 Rev 5 AC-17(2) | • Limit network access | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/WWSARQYKQSTH |
|                     | NIST 800-53 Rev 5 AC-4 | | • US East (Ohio)  
arn:aws:controltower:us-east-2::control/XRLOSBAPFQJX |
|                     | NIST 800-53 Rev 5 IA-5(1) | | • US West (Oregon)  
arn:aws:controltower:us-west-2::control/UBFDEGMVXN1 |
|                     | NIST 800-53 Rev 5 SC-13 | | • Canada (Central)  
arn:aws:controltower:ca-central-1::control/RZQGPORFMUYS |
|                     | NIST 800-53 Rev 5 SC-23(3) | | • Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/OLUIOTBOOCS |
|                     | NIST 800-53 Rev 5 SC-7(4) | | • Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/CXHKQ1PMPPO |
<p>|                     | NIST 800-53 Rev 5 SC-8 | | |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/DFPIFORHUVAL</td>
</tr>
<tr>
<td>2.3</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ZSEFDTLWVOAK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/JOEBOYAVBHVM</td>
</tr>
<tr>
<td>4.1</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/PPZGFKMRMNY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/JICDLDNZKCWO</td>
</tr>
<tr>
<td>8.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/CCEBBTRLFTN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/EWSEAKUVMBGZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/IXOEZMSDXZWD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/KALMFRNXCPDH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/WOMICOLFRVLF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/UDCLC3O0JKFE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Jakarta) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>southeast-3::control/ASSEXCKVIKWR</td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/NSGUUZCKLJMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/SNQRKWNNHOR</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/VUSTFQGUNEMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/LWBRONIIXUIV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/VYZZGPFTFEI</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/QXAQEWRRJRAG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/VYRQOKMPEK</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/CSEUEGCRHYO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/XKYYXMSPCZY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/LMBAWXBHTSZN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.ELASTICLOADBALANCING.PR.13 | NIST 800-53 Rev 5 | • Improve availability | • US East (N. Virginia) 
arn:aws:controltower:us-east-1::control/YYKNHJ6GCTZX  
US East (Ohio) 
arn:aws:controltower:us-east-2::control/ARGINPIXWQOF  
US West (Oregon) 
arn:aws:controltower:us-west-2::control/ELKXDUVDUCCS  
Canada (Central) 
arn:aws:controltower:ca-central-1::control/SHNYFGRZOKNR  
Asia Pacific (Sydney) 
arn:aws:controltower:ap-southeast-2::control/VDPRZIODKYSS  
Asia Pacific (Singapore) 
arn:aws:controltower:ap-southeast-1::control/WQMJLMPNJRHT  
Europe (Frankfurt) 
arn:aws:controltower:eu-central-1::control/PDRQZUMHBWIZ  
Europe (Ireland) 
arn:aws:controltower:eu-west-1::control/DWLXBJDQNJFH  
Europe (London) 
arn:aws:controltower:eu-west-2::control/IEHVGTXGXGN  
Europe (Stockholm) 
arn:aws:controltower:eu-north-1::control/MXZLYVQILODC  
Asia Pacific (Mumbai) 
arn:aws:controltower:ap-south-1::control/LHLSDALYHRNL |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/YAKLMULFRDKX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/KOHVKYUF PXUZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/GQXWFKUIIYZA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/QZHVAVUVBIYQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/QDFVSDJSXUNH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/XNMLHOXTXTYZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/OOROR GRUVA YIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ITCUUSRTKCPQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/ORVXQIODQZ QI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/DKSVEZCPPUAO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/VEJXGZNVVZTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.ELASTICLOADBALANCING.PR.14</td>
<td>NIST 800-53 Rev 5 CP-10</td>
<td>Improve resiliency, Improve availability</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/VCZFJEAUAWLN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/JLZRGIKJAFUW</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-36</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/JAZZGQBQ8ZO</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:can-central-1::control/MEFQWCBRO0NN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>southeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIBTD0HYUMUL</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Singapore) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>southeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WMYUGBVGEVD</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MHMICTUWLSON</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WBPOTRYBSJLHO</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NBZVVOENITM</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DMQFZQBAOKW</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Mumbai) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ORKESKWKXKFS3</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ASEAWQQCERPP</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Tokyo) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>northeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KOYYTIUQCFMH</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ZTOHLMITRHNW</td>
</tr>
<tr>
<td>• South America</td>
<td></td>
<td></td>
<td>(São Paulo) arn:aws:controltower:sa-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TDECASRAMXGX</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td>California) arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PZTKTHNYEPU</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-east-1::control/LICKEJRNYWTG</code></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-southeast-3::control/AKZIMZMBHXPK</code></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-northeast-3::control/OBHPNWWYXELNI</code></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:eu-south-1::control/REZXCUMITGVU</code></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:af-south-1::control/NVGYXJLZLG</code></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:me-south-1::control/DZKOPKXFMNZV</code></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:il-central-1::control/KJCDHGDVGD</code></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:eu-central-2::control/SWBFYNAEJPIC</code></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:eu-south-2::control/ZVFSRQSYLXG</code></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-south-2::control/NYLPBEXSGGOW</code></td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:me-central-1::control/YLDIOIDHZXEX</code></td>
</tr>
<tr>
<td>• Asia Pacific (Melbourne)</td>
<td></td>
<td></td>
<td><code>arn:aws:controltower:ap-</code></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.ELASTICLOADBALANCING.PR.15</td>
<td>NIST 800-53 Rev 5 CP-10</td>
<td>• Improve availability</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ESMQBBSNPPBL</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/DWNVOOZNWBZK</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/SFJRXHPEEWFA</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/WAIVIZIPAGHI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/RTXCNWMADBTT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/NKWJBUZINNH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/FGLPHBUZPHIK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/CLRMSTTALUWI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ITVPCOCJZHLR</td>
</tr>
</tbody>
</table>

CT.ELASTICLOADBALANCING.PR.15
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/SFH5NOQUTAHQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/QWLT0TPYPBRS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/NCOYGDOCXNGL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/RQYJFNYNJJUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/YSPLCVNCDBH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/QNQCBDIMDQUG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/DBLEVFJHLDwB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ZALYTABUTANC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/TRVJVOOTGFUQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/OCYADTLKTJSO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/RLOXGPMIMCHHC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.ELASTICLOADBALANCING.PR.2</td>
<td>NIST 800-53 Rev 5 AC-17(2)</td>
<td>• Encrypt data in transit</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/TWN0USHMRFWV</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/XDTTHNUPIHNG</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 IA-5(1)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/KPAIALQFVPZD</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-12(3)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-23</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-23(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-23(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **central-1::control/**
  - UQPWYSTFWPNX
- **Asia Pacific (Sydney)**
  - arn:aws:controltower:ap-southeast-2::control/OPRFIUYZWTCD
- **Asia Pacific (Singapore)**
  - arn:aws:controltower:ap-southeast-1::control/IBOADPMIQPQQ
- **Europe (Frankfurt)**
  - arn:aws:controltower:eu-central-1::control/QGSARYANMWWYN
- **Europe (Ireland)**
  - arn:aws:controltower:eu-west-1::control/UXJGSEXNOJMU
- **Europe (London)**
  - arn:aws:controltower:eu-west-2::control/SYMCEDHRRDZWX
- **Europe (Stockholm)**
  - arn:aws:controltower:eu-north-1::control/WIFXFDAZYLAY
- **Asia Pacific (Mumbai)**
  - arn:aws:controltower:ap-south-1::control/YSAHARNPJCBU
- **Asia Pacific (Seoul)**
  - arn:aws:controltower:ap-northeast-2::control/NZJCESRANJAKF
- **Asia Pacific (Tokyo)**
  - arn:aws:controltower:ap-northeast-1::control/TNEWNBTNWBIX
- **Europe (Paris)**
  - arn:aws:controltower:eu-west-3::control/KZUEDDHPVJPMAE
- **South America (São Paulo)**
  - arn:aws:controltower:saeast-1::control/0GNNHFZTHGDA
- **US West (N. California)**

---

1872
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>arn:aws:controltower:us-west-1::control/HVJNQRHVMBMZ</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/TKICAYPVLEWI</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-southeast-3::control/FTFRBHJUCHHK</td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-northeast-3::control/YNXDMUSTJ</td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-1::control/TCAIAXDVEICZ</td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eusouth-1::control/UEYBXXZRGQHS</td>
</tr>
<tr>
<td>arn:aws:controltower:af-south-1::control/0GBTHHEOFJIM</td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/WIBUNBPATHSS</td>
</tr>
<tr>
<td>arn:aws:controltower:me-central-1::control/XSAGNUBEYFF</td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-central-1::control/WIBUNBPATHSS</td>
</tr>
<tr>
<td>arn:aws:controltower:il-central-1::control/WIBUNBPATHSS</td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/WIBUNBPATHSS</td>
</tr>
<tr>
<td>arn:aws:controltower:eucentral-2::control/UEYBXXZRGQHS</td>
<td></td>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eusouth-2::control/IDEVSZEMXXFJ</td>
</tr>
<tr>
<td>arn:aws:controltower:eusouth-2::control/IDEVSZEMXXFJ</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/BJCGMXBUQOAS</td>
</tr>
<tr>
<td>arn:aws:controltower:me-central-1::control/APWNTETPZCJN</td>
<td></td>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/APWNTETPZCJN</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.ELASTICLOADBALANCING.PR.3 | NIST 800-53 Rev 5 AC-4(21) | • Protect data integrity | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/VLCKSWJHUUNN  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/WORKYXIAEPPY  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/RQXEJKEEBQR  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/BLKONTFMJJIM  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/DTFRJGDGCVDJ  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/OHOVDEGRDBJY  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/EBWHWDARURZT  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/EDDMFEEESKTR  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/GY LTSXUFSDYJ | NIST 800-53 Rev 5 CA-9(1)  
NIST 800-53 Rev 5 CM-2  
PCI DSS version 3.2.1  
1.2.1  
PCI DSS version 3.2.1  
1.3  
PCI DSS version 3.2.1  
1.3.1  
PCI DSS version 3.2.1  
1.3.2  
PCI DSS version 3.2.1  
1.3.4  
PCI DSS version 3.2.1  
2.2.3  
PCI DSS version 3.2.1  
6.6 |  | • Asia Pacific  
(Melbourne)  
arn:aws:controltower:ap-southeast-4::control/BVFELBGGGJLE |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/VMKVMHBBXJXT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/UQNSEXWGOAVI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/DZKUXADKBLK0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/SLYLLBVCPSQB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/QRQEUQSJGTQX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/INC300CTGBSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/RTMVFSWFTQBU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/CUUSDQGUOMWE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/BMTMYGwKGRT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/IVJMHGRXQFTX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/0IGJSCKAHJKA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.ELASTICLOADBALANCING.PR.4</td>
<td></td>
<td>Protect configurations</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ZVUGGRKPSNF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/GIZ0ETCRS3GJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-1876</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>west-2::control/CZGWNMZGUQQG</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/GPGUBQHOYWEA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/REBMKCJKOTPA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/GGMITQUCXRC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/AYTYFKBZPPNU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/STVTFNNQGFNC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/TNP1GQODBEEP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/HCRXAYUXHXVH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/YXLUVVLBYMP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/GMFGDGANQPXL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/WJEXQWQTKVUN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/AIVQIPFVIVA</td>
</tr>
</tbody>
</table>
| | | | South America (São Paulo) arn:aws:controltower:sa-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>MRCIZCRRATWL</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>U00YGDJUCEQK</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ZHTQABLCBMIJ</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JLDJRXDHOC5</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LSUGITKOSTL8</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GFUUYAYTPGJL</td>
</tr>
<tr>
<td>• Africa (Cape</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WDLCZEEWASP</td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>YII0HKYEOXLP</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PDPKRTBURTGG</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:europe-central-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TOIlKTMTPUM</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>XVSQHWPOKDLH</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Hyderabad)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GCBLJGHIUUAX</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.ELASTICLOADBALANCING.PR.5</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Improve availability</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/XZMJRFQFAFBQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2(2)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/VMTTTZPRYKXJ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:central-1::control/LZTVNXQBDIDF</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/RNHSNSUAJPAP</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/FCBAHAOWAPAG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eucentral-1::control/LWERVUDADSHP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eucentral-1::control/HZMwWCHIOPKU</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/CZOZUMCNAXGV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/TQLTEQM3SACM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/NFDTXLIHKABM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/KZPLSIUXH2LU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/JGGUSYGAEZ0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/VZJKBZOLBBES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/YEXUASWESFBT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/XCFEFIVKEHSY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/KLKOLHYUIDKN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/PFRMCVVXGCWD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/REBZVFIVNNZM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Europe (Milan)| arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ELASTICLOADBALANCING.PR.6</td>
<td>NIST 800-53 Rev 5 AC-4(26)</td>
<td>• Establish logging and monitoring</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ACYJDQTSIBRW</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-1::control/ACYJDQTSIBRW</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td>• US Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/KGOLLBSPLUFJR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/UQJJGFUCDAOAL</td>
</tr>
</tbody>
</table>

- **Framework**: NIST 800-53 Rev 5
- **Control objective**: Establish logging and monitoring
- **Control API identifiers, by Region**:
  - **US East (N. Virginia)**: arn:aws:controltower:us-east-1::control/ACYJDQTSIBRW
  - **US East (Ohio)**: arn:aws:controltower:us-east-1::control/ACYJDQTSIBRW
  - **US Pacific (Hyderabad)**: arn:aws:controltower:ap-south-2::control/KGOLLBSPLUFJR
  - **US Pacific (Melbourne)**: arn:aws:controltower:ap-southeast-4::control/UQJJGFUCDAOAL
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td></td>
<td>east-2::control/FJIEEFHBKMEI</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/KBBQEOOHAYIT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/UHDFKBXKLD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/TGQXQXZCCESS</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/UGTFHAIJCBE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/CXDFUAKPJF</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/TIGQHUXHQKW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/PQECPEVQHKU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/JOHHSDFRJVXT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/FKDCIAVBJHLD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/YQZXXORNYIQK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/JFIXYOKMPDZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective, by Region</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------------------</td>
<td></td>
</tr>
<tr>
<td>west-3::control/AOXHNAFCXEQZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:sa-east-1::control/JTVDNERKQMB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/IASBZFRYDXTB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-east-1::control/DISFRWVXQKSD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-southeast-3::control/BCCTQMAXUQQP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:ap-northeast-3::control/LHRLWZAGTIE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-1::control/MRIJZHGYXLT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:af-south-1::control/LNOAJYUTFFUA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:me-south-1::control/MQNFQYDACXIF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:il-central-1::control/VYKHSDEISAB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-central-2::control/IYMNAMMHPHC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-2::control/YEQCNPDRFTF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.ELASTICLOADBALANCING.PR.7</td>
<td>NIST 800-53 Rev 5 CP-10</td>
<td>Improve availability</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/CWXBBUJAKHIY</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/JQWTMKADKIQM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/RXBCLXMNDYLS</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/FSRKRIKNTVD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/OETUGRLWDUUA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/IAJRQALUGLVC</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/PMUEIFETYEVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/SEDMCNWATELI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/UNYSAQENBDWHP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/RSTFVKGDFFVP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/PNSQXKURZA00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ERFOIJBAKGVW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/VANTXKGTIFYIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/YJOPKSHLSHDU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/NIYWGCEVFEP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/JGNRBPRFJUHZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/INBVXUZPITIY</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Asia Pacific (Osaka) | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.ELASTICLOADBALANCING.PR.8</td>
<td>NIST 800-53 Rev 5 AC-17(2)</td>
<td>Encrypt data in transit</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-central-1::control/</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
<td>east-1::control/</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 IA-5(1)</td>
<td></td>
<td></td>
<td>QQYKDUWYE0IM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-12(3)</td>
<td></td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-23</td>
<td></td>
<td></td>
<td>EAST-2::CONTROL/</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-23(5)</td>
<td></td>
<td></td>
<td>NYJAEPPDXXSFP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td></td>
<td>ARN:AWS:CONTROLLTOWER:US</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(1)</td>
<td></td>
<td></td>
<td>WEST-2::CONTROL/</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td></td>
<td>HDFFHQGMCHAT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3</td>
<td></td>
<td></td>
<td>ARN:AWS:CONTROLLTOWER:CA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td>CENTRAL-1::CONTROL/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>NHXVNOSXDCMRQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>ARN:AWS:CONTROLLTOWER:AP</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>SOUTHEAST-1::CONTROL/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>ASIA PACIFIC-1::CONTROL/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>AP SOUTHEAST-2::CONTROL/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>ASIA PACIFIC-2::CONTROL/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>EUROPE FRANKFURT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>ARN:AWS:CONTROLLTOWER:EU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>CENTRAL-1::CONTROL/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>NSQJFFBDYFJE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>ARN:AWS:CONTROLLTOWER:EU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>WEST-2::CONTROL/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>HQHFHTSBARTW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>ARN:AWS:CONTROLLTOWER:EU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>NORTH-1::CONTROL/</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>QBKEGNOQIMOWD</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>ARN:AWS:CONTROLLTOWER:AP</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>SOUTH-1::CONTROL/</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>JDAURPDZMOS</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>ARN:AWS:CONTROLLTOWER:AP</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>NORTHEAST-2::CONTROL/</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>ASIA PACIFIC (Tokyo)</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>ARN:AWS:CONTROLLTOWER:AP</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>EU</td>
<td></td>
<td></td>
<td>northeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TUPOKAOWCEWX</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>• arn:aws:controltower:eu-west-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CRJEOQPBHUYN</td>
</tr>
<tr>
<td>South America</td>
<td></td>
<td></td>
<td>• arn:aws:controltower:sa-east-1::control/</td>
</tr>
<tr>
<td>(São Paulo)</td>
<td></td>
<td></td>
<td>EEDDEBDDSZHT</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>• arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LXKUWICNDPUQ</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td></td>
<td></td>
<td>• arn:aws:controltower:ap-east-1::control/</td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td></td>
<td></td>
<td>TUAETWCFNPHP</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>• arn:aws:controltower:ap-southeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HLJJOXOYQRW0</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>• arn:aws:controltower:ap-northeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>QYYUHUZOKXXSI</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>• arn:aws:controltower:eu-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BIWEKWO3FQPY</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
<td>• arn:aws:controltower:af-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TLBPPGCCHPWD</td>
</tr>
<tr>
<td>Middle East</td>
<td></td>
<td></td>
<td>• arn:aws:controltower:me-south-1::control/</td>
</tr>
<tr>
<td>(Bahrain)</td>
<td></td>
<td></td>
<td>SSKNKBUIWGWG</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>• arn:aws:controltower:il-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>XPTUPNEMJDRJ</td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td></td>
<td></td>
<td>• arn:aws:controltower:eu-central-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HCENLRGKUPQG</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.ELASTICLOADBALANCING.PR.9 | NIST 800-53 Rev 5 | Encrypt data in transit | • Europe (Spain)  
  arn:aws:controltower:eu-south-2::control/GNGTFFXVJZOQ  
  • Asia Pacific (Hyderabad)  
  arn:aws:controltower:ap-south-2::control/K3LVwZWYXECE  
  • Middle East (UAE)  
  arn:aws:controltower:me-central-1::control/EISTKDGVFVYA  
  • Asia Pacific (Melbourne)  
  arn:aws:controltower:ap-southeast-4::control/JLJILSWJBAWD  

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.ELASTICLOADBALANCING.PR.9 | NIST 800-53 Rev 5 AC-17(2) | Encrypt data in transit | • US East (N. Virginia)  
  arn:aws:controltower:us-east-1::control/OJRPUPOEBOGC  
  • US East (Ohio)  
  arn:aws:controltower:us-east-2::control/BZAIVZY1G5BEA  
  • US West (Oregon)  
  arn:aws:controltower:us-west-2::control/DYNNAOSHIGUA  
  • Canada (Central)  
  arn:aws:controltower:ca-central-1::control/ZFDYMUBADTU  
  • Asia Pacific (Sydney)  
  arn:aws:controltower:ap-southeast-2::control/XCEAB5SKXHNEW  
  • Asia Pacific (Singapore)  
  arn:aws:controltower:ap-southeast-1::control/ONUHDZQBRTPI |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 2.3</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/EWMJJYUDXREQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/WQWNYGWHUHDW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/VEQTOWINFEKD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/STGPISYTFYUO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/RGANMYVCMWFGZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/LYLRCNQCPNJV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/GKAFFTUBEDGA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/TEDBEMVAAJTO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/QQZDUTOOGNKG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/BKNZWUBNRNNK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/DLTVTIXROSGT</td>
</tr>
</tbody>
</table>
| | | | • Asia Pacific (Jakarta) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>southeast-3::control/FAKSKCVVGLZ</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/CTMACEMAYWWP</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/IZWNTXSZPIHY</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/XFQFMXGYKBNP</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/XOVPGIDDDPA</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/KDVMCUOKDUDK</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/XKGBOMUJQHTQ</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/AMSZOJWMYPMVA</td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/TRLOIHFYIRMI</td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/MZHSLHDOAIME</td>
</tr>
<tr>
<td>• Asia Pacific (Melbourne)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/LFVOMXNGCJAX</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
</tbody>
</table>
| CT.EMR.PR.1        | • NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-3(6)  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-28  
• NIST 800-53 Rev 5 SC-28(1)  
• NIST 800-53 Rev 5 SC-7(10)  
• NIST 800-53 Rev 5 SI-7(6)  
• PCI DSS version 3.2.1  
10.5  
• PCI DSS version 3.2.1  
10.5.2  
• PCI DSS version 3.2.1  
2.2  
• PCI DSS version 3.2.1  
3.4 | • Encrypt data at rest | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
CGKQCHLVDEIB  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
YHCUCJSGQFXA  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
IVSACGXXHRSR  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
INXDKMEOBTF  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
TFCTANMLNGMR  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
KSGLCSMAEEDA  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/  
MMPGM1GJQHON  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/  
TWSLMQLEOHKS  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/  
KEGOEDVBGLRF  
• Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/  
QSKWRYDXXVED  
• Asia Pacific (Mumbai)  
arn:aws:controltower:ap-south-1::control/  
TVAYHGOUXLPE |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/ESGKLHSP0DEY</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/CTSXXPDCUEDR</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/KTMOCNTSAXOL</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/GZCNFBZSQDOQ</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/UFVDCAJPRVWU</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/MBPQYGMWWUN</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/HGZIJECVJGD</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/OMAMQJBXCRMU</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/RJOLQVEUPW</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/SAPJRYGDPIJG</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/BPFAHEYTAYOS</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.EMR.PR.2</td>
<td></td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia)\narn:aws:controltower:us-east-1::control/\nRBBMAHVKFUYB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio)\narn:aws:controltower:us-east-2::control/\nUNMATKWGXVEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon)\narn:aws:controltower:us-west-2::control/\nTRYHGCGPXQBY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central)\narn:aws:controltower:ca-central-1::control/\nQSNQBMVQI0GP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)\narn:aws:controltower:ap-south-4::control/\nFBOBOZDTKTRB</td>
</tr>
</tbody>
</table>

- **Europe (Zurich)**\narbn:aws:controltower:eu-central-2::control/\nAPMARIGMLCIH
- **Europe (Spain)**\narbn:aws:controltower:eu-south-2::control/\nHJIMSQRPTZSY
- **Asia Pacific (Hyderabad)**\narbn:aws:controltower:ap-south-2::control/\nLMBWEFMCQIYN
- **Middle East (UAE)**\narbn:aws:controltower:me-central-1::control/\nASFOABAEKFKU
- **Asia Pacific (Melbourne)**\narbn:aws:controltower:ap-southeast-4::control/\nFBOBOZDTKTRB

- **US East (N. Virginia)**\narbn:aws:controltower:us-east-1::control/\nRBBMAHVKFUYB
- **US East (Ohio)**\narbn:aws:controltower:us-east-2::control/\nUNMATKWGXVEF
- **US West (Oregon)**\narbn:aws:controltower:us-west-2::control/\nTRYHGCGPXQBY
- **Canada (Central)**\narbn:aws:controltower:ca-central-1::control/\nQSNQBMVQI0GP
- **Asia Pacific (Sydney)**\narbn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>southeast-2::control/CHDEFRRDBBNLC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/PFAXXMHSTOJG</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/SHPWLBFPVMVD</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/SFHGSRHYHWR</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/YWOIDYXYVQUE</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/VWUF6PVIPUEUJ</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/PHHKZGEWDAZK</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/NXJYKOBPZCEF</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/DDJLGXQNVUUQ</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/OPAFIWVUBIIF</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:saeast-1::control/WFNNKLCKRXJE</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/NPCHIRMROYBBA</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/QHFQNHUAJJOS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/PIXVJDFYLEEK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/EBFHBZHPBGVB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/WSQGBUDGZQBD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/MNUYLLRYYTHP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/FBIJDMIEXC0G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/JDVJIYRFXULB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/NVJSHTUXXIIY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/DANDKACRGMEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/FHSOUQJENGFW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/JLKIPCOHILYC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Asia Pacific (Melbourne) | arn:aws:controltower:ap-
## CT.EMR.PR.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.EMR.PR.3</td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Encrypt data at rest</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/QNPZCJDNGGSI</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/PILEMYYHTDXA</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/UDEMB6IGVWRQ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/UDKRPGZVFDUE</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/PMKIZZXISXGC</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/LJLBK003FQZD</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/NUJINJQLQ0UN</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.5</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/JAHQGPNBHEJI</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.5.2</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/UPFHHKEZY0CC</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/OCDDXWMLCWVT</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective, by Region</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>----------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>south-1::control/XSTLMPKVTZP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>northeast-2::control/YIITJQP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>northeast-1::control/PRNNBEUWYTVA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>west-3::control/QVNKNUJGSZID</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• South America (São Paulo)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>east-1::control/SIAVBUUYLWBK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• US West (N. California)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:us-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>west-1::control/PBZZRBLAUMUO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>east-1::control/QYNTMPMVILFC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>southeast-3::control/GMWRLUZJKUV</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>northeast-3::control/LVBHITUYTG</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>south-1::control/IEISUVEELSRQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:af-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>south-1::control/MVGLRQDIGVKY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>arn:aws:controltower:me-</td>
<td></td>
</tr>
</tbody>
</table>
## Control identifier CT.EMR.PR.4

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.EMR.PR.4        |           | Encrypt data in transit | US East (N. Virginia)  
  arn:aws:controltower:us-east-1::control/VKFEQPOWCEQO  
  US East (Ohio)  
  arn:aws:controltower:us-east-2::control/EHLZOFIPQELW  
  US West (Oregon)  
  arn:aws:controltower:us-west-2::control/GFUOQRPOJQO  
  Canada (Central)  
  arn:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>central-1::control/PBXJBZPNELGX</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/YLGTYOLGXBHH</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/IPWAJHVDSLKX</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eucentral-1::control/JCKGMJ0AJMOP</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:euwest-1::control/TETBYMUVVMCK</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:euwest-2::control/NGFBINLXWMVZV</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eunorth-1::control/OMIMVRHODCMR</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:apsouth-1::control/HIZUGNMVVFIV</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:apnortheast-2::control/RDHYEEFAEUMS</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:apnortheast-1::control/ISRNTOKYVYFL</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:euwest-3::control/TQDRKJ0STGF</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/IWHTHELJEWQCQ5</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/CRNDYYYDDGZL</td>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/UHPYOGSODQAI</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-east-1::control/UHPYOGSODQAI</td>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/NWZAQBRTRWQ</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-southeast-3::control/NWZAQBRTRWQ</td>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/V51000CSNCCXV</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-northeast-3::control/V51000CSNCCXV</td>
<td>• Europe (Milan)</td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/GKMOMEBKHZEL</td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-1::control/GKMOMEBKHZEL</td>
<td>• Africa (Cape Town)</td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/WHAGYFSPKXE</td>
</tr>
<tr>
<td>arn:aws:controltower:af-south-1::control/WHAGYFSPKXE</td>
<td>• Middle East (Bahrain)</td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/YXZUTBJKJAI</td>
</tr>
<tr>
<td>arn:aws:controltower:me-south-1::control/YXZUTBJKJAI</td>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/NHRUSBNKRQBU</td>
</tr>
<tr>
<td>arn:aws:controltower:il-central-1::control/NHRUSBNKRQBU</td>
<td>• Europe (Zurich)</td>
<td></td>
<td>arn:aws:controltower:eucentral-2::control/GUJOBBDBOIBJ</td>
</tr>
<tr>
<td>arn:aws:controltower:eucentral-2::control/GUJOBBDBOIBJ</td>
<td>• Europe (Spain)</td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/LXJWHUCUNCCF</td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-2::control/LXJWHUCUNCCF</td>
<td>• Asia Pacific (Hyderabad)</td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/PRXVTLEHHJRV</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-south-2::control/PRXVTLEHHJRV</td>
<td>• Middle East (UAE)</td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/XVYHZAEAOUWJ</td>
</tr>
<tr>
<td>arn:aws:controltower:me-central-1::control/XVYHZAEAOUWJ</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CT.GLUE.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.GLUE.PR.1       | • NIST 800-53 Rev 5 CA-9(1)  
                     • NIST 800-53 Rev 5 CM-3(6)  
                     • NIST 800-53 Rev 5 SC-13  
                     • NIST 800-53 Rev 5 SC-28  
                     • NIST 800-53 Rev 5 SC-28(1)  
                     • NIST 800-53 Rev 5 SC-7(10)  
                     • NIST 800-53 Rev 5 SI-7(6)  
                     • PCI DSS version 3.2.1 10.5  
                     • PCI DSS version 3.2.1 10.5.2  
                     • PCI DSS version 3.2.1 2.2  
                     • PCI DSS version 3.2.1 3.4 | • Encrypt data at rest | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
WPCRZFUMPUYK  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
TCWRJOJCLVFQ  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
SKMTAZQLOSYY  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
IEKSUNOKGWLL  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
UETCETTAWIFA  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
QQFDELOYZHJ  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/  
ZABAPQCMENQX  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/  
GFTITBQSMXZN  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/  
PADEKLSUDPRA |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/LIJZZPPMMXINO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/TLZNZDRMXYUD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/FJGIMXRXTUMO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/LZYUTOYMDJEV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/ULXVMADMBHBW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/TLFCNIUBKRAX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/TTFHKBNXUHRD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/SVYDUSNGTBBAO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/OYFSLJFWANFX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/IOYSTBILSAIQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/FTVODADNCSEV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>CT.GLUE.PR.2</td>
<td></td>
<td></td>
<td>south-1::control/ JWHFNF3FLFRPL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/ ZYZYJEFMGLPZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/ TWMRRWAZ5H0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eucentral-2::control/ TQ0QWVNSWRP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eusouth-2::control/ WQYRTUZWXCT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:apsouth-2::control/ NZLHB3LAIORXT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:mecentral-1::control/ PJXYGBKLLGYYV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:apsoutheast-4::control/ VGEVD3HSNCJN</td>
</tr>
<tr>
<td>CT.GLUE.PR.2</td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Encrypt data at rest</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:useast-1::control/ GMDWIONESNXP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:useast-2::control/ IIFYNGDGKOBV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:uwest-1::control/ GMDWIONESNXP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:uwest-2::control/ IIFYNGDGKOBV</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **west-2::control/**
  - ZAYLOZUJAMZA
- **Canada (Central)**
  - arn:aws:controltower:ca-central-1::control/
  - FXDVFWNOGHTPT
- **Asia Pacific (Sydney)**
  - arn:aws:controltower:ap-southeast-2::control/
  - UCEYHTHJRXVY
- **Asia Pacific (Singapore)**
  - arn:aws:controltower:ap-southeast-1::control/
  - YMWNGPYLHJJD
- **Europe (Frankfurt)**
  - arn:aws:controltower:eu-central-1::control/
  - ACSFCDUZGQFR
- **Europe (Ireland)**
  - arn:aws:controltower:eu-west-1::control/
  - RKCBM3GPUCAK
- **Europe (London)**
  - arn:aws:controltower:eu-west-2::control/
  - QGHHRCZKXSXY
- **Europe (Stockholm)**
  - arn:aws:controltower:eu-north-1::control/
  - BFWOFLEWPFTD
- **Asia Pacific (Mumbai)**
  - arn:aws:controltower:ap-south-1::control/
  - DUEWFGEYHTFR
- **Asia Pacific (Seoul)**
  - arn:aws:controltower:ap-northeast-2::control/
  - CNYTCMUENMHU
- **Asia Pacific (Tokyo)**
  - arn:aws:controltower:ap-northeast-1::control/
  - DRLGLUTHSINIS
- **Europe (Paris)**
  - arn:aws:controltower:eu-west-3::control/
  - RYFQPUZXYXFR
- **South America (São Paulo)**
  - arn:aws:controltower:sa-

1905
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>QXTABOWUIJJK</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td></td>
<td>IUQBRTWTDCHC</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-east-1::control/</td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td></td>
<td></td>
<td>DAAUYHJNSNAL</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-southeast-3::control/</td>
</tr>
<tr>
<td>(Jakarta)</td>
<td></td>
<td></td>
<td>SGIFRDQDWSXZ</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-northeast-3::control/</td>
</tr>
<tr>
<td>(Osaka)</td>
<td></td>
<td></td>
<td>LHISQHTENKTI</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:eusouth-1::control/</td>
</tr>
<tr>
<td>• Africa (Cape</td>
<td></td>
<td></td>
<td>MXNOTFXWBRXF</td>
</tr>
<tr>
<td>Town)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:af-south-1::control/</td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td></td>
<td>SJSGPCULXYUK</td>
</tr>
<tr>
<td>(Bahrain)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:mesouth-1::control/</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>PGEUPSLHOPXR</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:euzentral-2::control/</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>TQFGPSUROWOK</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:eusouth-2::control/</td>
</tr>
<tr>
<td>(Hyderabad)</td>
<td></td>
<td></td>
<td>NQNRCWIPNBEI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-south-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>QNIFFKAFDJFQ</td>
</tr>
</tbody>
</table>
### CT.GUARDDUTY.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.GUARDDUTY.PR.1 | • NIST 800-53 Rev 5 AC-2(12)  
• NIST 800-53 Rev 5 AU-6(1)  
• NIST 800-53 Rev 5 AU-6(5)  
• NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 RA-3(4)  
• NIST 800-53 Rev 5 SC-7(10)  
• NIST 800-53 Rev 5 SI-4  
• NIST 800-53 Rev 5 SI-4(13)  
• NIST 800-53 Rev 5 SI-4(2)  
• NIST 800-53 Rev 5 SI-4(4)  
• PCI DSS version 3.2.1 11.4 | • Protect configurations  
• Prepare for incident response | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/HDYIPPMSW2OU  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/PYNXRWWBH6DCV  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/AAEWPWCHTL7BT  
• Canada (Central)  
ar:n:aws:controltower:ca-central-1::control/ETRMJIXAELLB  
• Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/CELJQOZUYTCC  
• Asia Pacific (Singapore)  
ar:n:aws:controltower:ap-southeast-1::control/YTVHWWKK3NVVZ  
• Europe (Frankfurt)  
ar:n:aws:controltower:eu-central-1::control/HFQIQZNQEDBF  
• Europe (Ireland)  
ar:n:aws:controltower:eu-west-1::control/QBUMAVYUWFG |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/EZEFIVKLTUWW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/WTYFKSYNHC0E</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/CGJWZYIKSAKC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/KWNBRWJUUAAK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/SPIMIVFZLKG7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/YGKBDKCNXMAS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/BUDEACEBNFPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/XGBAGKTSFTLVU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/KUKBAIDPCVEP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/QQXJTSRQRQ5IC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/GATXICTOAKVM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-</td>
</tr>
</tbody>
</table>
## Control identifier CT.IAM.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.IAM.PR.1        | • CIS AWS Benchmark 1.4 1.16  
                   |           | • Enforce least privilege | • US East (N. Virginia)  
                   |           |                                | arn:aws:controltower:us-east-1::control/  
                   |           |                                | JMBMAKMUHMKC |
|                    | • NIST 800-53 Rev 5  
                   |           |                                | • US East (Ohio)  
                   |           |                                | arn:aws:controltower:us-east-1::control/  
                   |           |                                | JMBMAKMUHMKC |

### Control objective
- Enforce least privilege
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>east-2::control/VRYGYLBERJDX</td>
</tr>
<tr>
<td>AC-2(1)</td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/IEYHGQBFVLSK</td>
</tr>
<tr>
<td>AC-3</td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/XSO DGKTQFXRF</td>
</tr>
<tr>
<td>AC-3(15)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/JOYWUUCESSYI</td>
</tr>
<tr>
<td>AC-3(7)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eucentral-1::control/VLBUBGROVKQD</td>
</tr>
<tr>
<td>AC-5</td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/OVZGSDCWZLCD</td>
</tr>
<tr>
<td>AC-6</td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/GXTMGHVRNDOR</td>
</tr>
<tr>
<td>AC-6(10)</td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/STCTJ1ILZFUDN</td>
</tr>
<tr>
<td>AC-6(2)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/UTUYMAXJR6ID</td>
</tr>
<tr>
<td>AC-6(3)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/AHDC TTXNKGUM</td>
</tr>
<tr>
<td>AC-6(10)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/AZWOFYWOCYFD</td>
</tr>
<tr>
<td>AC-6</td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>2.2</td>
<td></td>
<td>arn:aws:controltower:eucentral-1::control/JOYWUUCESSYI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.1.1</td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.1.2</td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/OVZGSDCWZLCD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.2.1</td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.2.2</td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/GXTMGHVRNDOR</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.2.3</td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.2.4</td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/STCTJ1ILZFUDN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.2.5</td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.2.6</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/UTUYMAXJR6ID</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.2.7</td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.2.8</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/AHDC TTXNKGUM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.2.9</td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.0</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/AZWOFYWOCYFD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.1</td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.2</td>
<td></td>
<td>arn:aws:controltower:eucentral-1::control/JOYWUUCESSYI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.3</td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.4</td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/OVZGSDCWZLCD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.5</td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.6</td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/GXTMGHVRNDOR</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.7</td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.8</td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/STCTJ1ILZFUDN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.9</td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.10</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/UTUYMAXJR6ID</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.11</td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.12</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/AHDC TTXNKGUM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.13</td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.14</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/AZWOFYWOCYFD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.15</td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.16</td>
<td></td>
<td>arn:aws:controltower:eucentral-1::control/JOYWUUCESSYI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.17</td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.18</td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/OVZGSDCWZLCD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.19</td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.20</td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/GXTMGHVRNDOR</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.21</td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.22</td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/STCTJ1ILZFUDN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.23</td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.24</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/UTUYMAXJR6ID</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.25</td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.26</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/AHDC TTXNKGUM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.27</td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.28</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/AZWOFYWOCYFD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.29</td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>7.3.30</td>
<td></td>
<td>arn:aws:controltower:eucentral-1::control/JOYWUUCESSYI</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>west-3::control/</td>
<td></td>
<td></td>
<td>WNQGLXPIXUGQTF</td>
</tr>
<tr>
<td>• South America</td>
<td></td>
<td></td>
<td>SJQPVXRJXEOUSF</td>
</tr>
<tr>
<td>(São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td>RKJQUGWSXPC</td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>LGJCRJPOVCTC</td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>ENDGXUBAFXAJ</td>
</tr>
<tr>
<td>(Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>ZHHFEZIYPBJK</td>
</tr>
<tr>
<td>(Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>ETMDQOKWHEREI</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/</td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td></td>
<td>WBJTGFVZTCA</td>
</tr>
<tr>
<td>(Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>DZOVLGFMXMI0</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------</td>
<td>-------------------</td>
<td>----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CT.IAM.PR.2</td>
<td>CIS AWS Benchmark 1.4 1.16</td>
<td>Enforce least privilege</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/PNOCTMYIRAQZ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-2(1)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/JMOGYFIPAVJA</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/HOKZSEQYEYHL</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/CAOCWBICFG0B</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/YHMPKYKTCUML</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-5</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/BJIKMVHRDDMG</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6(10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Framework</td>
<td>Control identifier</td>
<td>Control objective</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------</td>
<td>-------------------</td>
<td></td>
</tr>
</tbody>
</table>
| - PCI DSS version 3.2.1 7.2.1
  - PCI DSS version 3.2.1 7.2.2 | Europe (Ireland) | arn:aws:controltower:eu-west-1::control/ARQSBKHBXQCM |
|          | Europe (London)   | arn:aws:controltower:eu-west-2::control/ALIYKFTXVNCS |
|          | Europe (Stockholm)| arn:aws:controltower:eu-north-1::control/NFGGERHBCZUW |
|          | Asia Pacific (Mumbai)| arn:aws:controltower:ap-south-1::control/FEBWNSMFEJIB |
|          | Asia Pacific (Seoul)| arn:aws:controltower:ap-northeast-2::control/TRW CETXWEQUE |
|          | Asia Pacific (Tokyo)| arn:aws:controltower:ap-northeast-1::control/HOCOFZGQHTGD |
|          | Europe (Paris)    | arn:aws:controltower:eu-west-3::control/RAYNNMFDYEVC |
|          | South America (São Paulo)| arn:aws:controltower:sa-east-1::control/RSEEMCFURKTU |
|          | US West (N. California)| arn:aws:controltower:us-west-1::control/OGSIRAALHTWG |
|          | Asia Pacific (Hong Kong)| arn:aws:controltower:ap-east-1::control/CPYTOUZUVO MY |
|          | Asia Pacific (Jakarta)| arn:aws:controltower:ap-southeast-3::control/QI JWLSLISIUGXF |
|          | Asia Pacific (Osaka)| arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>

**CT.IAM.PR.3**

- **Control identifier**: CT.IAM.PR.3
- **Framework**: NIST 800-53 Rev 5 AC-2
- **Control objective**: Enforce least privilege
- **Control API identifiers, by Region**: US East (N. Virginia) arn:aws:controltower:us-1914
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td></td>
<td></td>
<td>east-1::control/XJWBVKPDALLC</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/UFUJIAVAHEMXT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/TQOQTVTFKYZP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/UVQFBXEVKVGX</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-5</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/DIYAWISHCSUP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/RBKJFTDCPGHB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(10)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/IRQIUXUHHNQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(2)</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/GDTCMKSZZJPE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(3)</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/DXWZZJDXVQJB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/YAYMMPOICSWS</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/UJVQWOKLAPQA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/SEDWSBCCUPUR</td>
</tr>
</tbody>
</table>
| • PCI DSS version 3.2.1 7.2.1 |  |  | • Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeas-1::control/</td>
<td></td>
<td></td>
<td>BHIAWIKNMSH</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:eu-west-3::control/ CXLKIUGCJVBV</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:saw-east-1::control/ YGJVNUUQETCS</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:us-west-1::control/ RJAICXMTHQFY</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-east-1::control/ ORPNELRUOFKT</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-southeast-3::control/ ILMQSXXDTGIW</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-northeast-3::control/ CJWYUKYSXKUP</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:eu-south-1::control/ SRCNCCOVAKTW</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:af-south-1::control/ XIIAJQOEISQW</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:me-south-1::control/ MBZAIPOCGKFG</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:il-central-1::control/ FYLPKJZTYMRN</td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:eu-central-2::control/ NDAJB3GDIAH</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>CT.IAM.PR.4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Control API identifiers, by Region**

- **Europe (Spain)**
  arn:aws:controltower:eu-south-2::control/HVIHRSXDN0GB

- **Asia Pacific (Hyderabad)**
  arn:aws:controltower:ap-south-2::control/LSOKRAEIVKHM

- **Middle East (UAE)**
  arn:aws:controltower:me-central-1::control/OBJIKKDCCRGV

- **Asia Pacific (Melbourne)**
  arn:aws:controltower:ap-southeast-4::control/RBTVVZELHWHT

- **US East (N. Virginia)**
  arn:aws:controltower:us-east-1::control/BGEGPSWZQGHE

- **US East (Ohio)**
  arn:aws:controltower:us-east-2::control/WDSXAJOAVHZA

- **US West (Oregon)**
  arn:aws:controltower:us-west-2::control/XGWPMANHZWI

- **Canada (Central)**
  arn:aws:controltower:ca-central-1::control/UEHHSMOZOLIQ

- **Asia Pacific (Sydney)**
  arn:aws:controltower:ap-southeast-2::control/N1QNDIAPPY1

- **Asia Pacific (Singapore)**
  arn:aws:controltower:ap-southeast-1::control/YXXSRJWMGBGN
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ALGNWNDJ0HGJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/HTVEKDWTMZF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/KNLJBRBNY00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/EYJVADFYATUK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/CIBVZXRPUAXJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/VPPG5MFNPEDX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ZTOG01LBNVLVZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/PAUXSALAWTJL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/ESBOQQWFA1JM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/ZCDLCRVFQOTU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/UFHNNLRVIPS</td>
</tr>
</tbody>
</table>
|                    |          |                  | • Asia Pacific (Jakarta) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>southeast-3::control/GQDQANYGJAPI</td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/SXUORJZQKEPJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/CNVGXHJPQJVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/CMHCTVLG0AZB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/WEIKDIOJBGFU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/ZGSPZACAZKYK</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/JGDUW0AAAFNAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/EHMJZNHWSSNR</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/JMWLMQKSOWRB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/NSSGQMAMTMLT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/SKIIBUQDKMNX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
</tbody>
</table>
| CT.IAM.PR.5         | • NIST 800-53 Rev 5 AC-2  
                      • NIST 800-53 Rev 5 AC-2(1)  
                      • NIST 800-53 Rev 5 AC-3  
                      • NIST 800-53 Rev 5 AC-3(15)  
                      • NIST 800-53 Rev 5 AC-3(7)  
                      • NIST 800-53 Rev 5 AC-5  
                      • NIST 800-53 Rev 5 AC-6  
                      • NIST 800-53 Rev 5 AC-6(10)  
                      • NIST 800-53 Rev 5 AC-6(2)  
                      • NIST 800-53 Rev 5 AC-6(3)  
                      • PCI DSS version 3.2.1 2.2  
                      • PCI DSS version 3.2.1 7.1.1  
                      • PCI DSS version 3.2.1 7.1.2  
                      • PCI DSS version 3.2.1 7.2.1  
                      • PCI DSS version 3.2.1 7.2.2 | • Enforce least privilege | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/CCMOVWIXXYLYF  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/ONHYGFQYMLZS  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/MCGPCOIVNQGM  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/CXYDXHZWZNXX  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/PSTEEFABICWJ  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/LZSJWCBDBCK  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/NSLUIOJZYNZS  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/IHTJEPISWWWF  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/TKRPZKLSOB  
• Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/WYOBTTMTGJXG  
• Asia Pacific (Mumbai)  
arn:aws:controltower:ap-south-1::control/JZHXBIKWJL
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/BXCZTL0DIJFG</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/ISCNDFSMVEVM</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/PRBDWMDNLJU</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/SUVBXVFLNFS</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/AUPBSLCCCX0KW</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/IZTCZMFKNZ0</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/HVFUXHOUQXP8</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/DOVTCMZPYCON</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/MISFEMWXCCG6</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/PYKMTSTZYDBO</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/JXNXTZPYKLI</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.KINESIS.PR.1</td>
<td></td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/HUIOKNICCRBB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ULXLVVUXLRR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/KHBDMVKH0BTL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/GJH1YBNTIXA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-4::control/QJVBMTNZMNYM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT.KINESIS.PR.1</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/HUIOKNICCRBB</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3(6)</td>
<td>Encrypt data at rest</td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ULXLVVUXLRR</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-13</td>
<td>Encrypt data at rest</td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/KHBDMVKH0BTL</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28</td>
<td>Encrypt data at rest</td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/GJH1YBNTIXA</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28(1)</td>
<td>Encrypt data at rest</td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-4::control/QJVBMTNZMNYM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(10)</td>
<td>Encrypt data at rest</td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-4::control/QJVBMTNZMNYM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-7(6)</td>
<td>Encrypt data at rest</td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-4::control/QJVBMTNZMNYM</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 3.4</td>
<td>Encrypt data at rest</td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-4::control/QJVBMTNZMNYM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypt data at rest</td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-4::control/QJVBMTNZMNYM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypt data at rest</td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-4::control/QJVBMTNZMNYM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypt data at rest</td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-4::control/QJVBMTNZMNYM</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>southease-2::control/KREYISSWDIIL</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/VPOCRYMFLHLQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/XCIQO5RDFVAR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/TCPBIRILCAQQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/CODJWYQCOXV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/IMXQJNMZ2J2AO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/QEXY5DBEDVZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/CCUPDGNBQPGW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/STVPVNMIFLX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/YANNDPDJPDKT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/XRDFTFPDNALI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/IQSVGYP0CMQQ</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/CPXPGDDIBIQV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/PSNLRLRRYHN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/CXQVUZQZUGDS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/IJPNWGFQMFNR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/KIPNHGILDQWA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/DASBBTCYBUOF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/HSTFUNSHQGEJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/MWRDPWNGLTZR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/OYCWERWIIYAE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/NYCMwP0JWLVZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/VOYLIBFBKLUN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Asia Pacific (Melbourne) | arn:aws:controltower:ap-
### CT.KMS.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.KMS.PR.1        | • CIS AWS Benchmark 1.4 3.8  
• NIST 800-53 Rev 5 SC-12  
• NIST 800-53 Rev 5 SC-12(2)  
• NIST 800-53 Rev 5 SC-28(3)  
• PCI DSS version 3.2.1 2.2  
• PCI DSS version 3.2.1 3.6.4 | • Encrypt data at rest | • US East (N. Virginia)  
ar:aws:controltower:us-east-1::control/VULBAJMYDWHA  
• US East (Ohio)  
ar:aws:controltower:us-east-2::control/JDPLYYQWRCOA  
• US West (Oregon)  
ar:aws:controltower:us-west-2::control/IJH0OMRUOCFD  
• Canada (Central)  
ar:aws:controltower:ca-central-1::control/TDLOCMNRYDEW  
• Asia Pacific (Sydney)  
ar:aws:controltower:ap-southeast-2::control/KZD5WGCNWWVI  
• Asia Pacific (Singapore)  
ar:aws:controltower:ap-southeast-1::control/FZQEVKYAAZWT  
• Europe (Frankfurt)  
ar:aws:controltower:eu-central-1::control/JKUXPIAYFKSK  
• Europe (Ireland)  
ar:aws:controltower:eu-west-1::control/VHIEQHFPSSHXY  
• Europe (London)  
ar:aws:controltower:eu-west-2::control/DEPFJLBPJFHZ  
• Europe (Stockholm)  
ar:aws:controltower:eu-north-1::control/YVFY1EHQZEWW
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/XSHCFZKRRKS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/FSQBJXDJBZVU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/KBUBBEPVJVA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/AOQXVQUAUJR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/SROYFMNTJWNJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/QZRDNMPHLQSS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/RBOOWYIGHTIP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/TLIKROYMIVQB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/RWZMAUHNERFS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/IQJLHCNTBZQA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/NIKMXFVUJCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CT.KMS.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.KMS.PR.2        | • NIST 800-53 Rev 5 CM-8  
  • PCI DSS version 3.2.1 2.2 | • Encrypt data at rest            | • US East (N. Virginia)  
  • US East (Ohio)  
  • US West (Oregon)  
  • Canada (Central) |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| central-1::control/UTUHYVIWSZJG  
- Asia Pacific (Sydney)  
  arn:aws:controltower:ap-southeast-2::control/NBBUFFKZOIWG  
- Asia Pacific (Singapore)  
  arn:aws:controltower:ap-southeast-1::control/CWZNIGTYZZAG  
- Europe (Frankfurt)  
  arn:aws:controltower:eu-central-1::control/BSVBULVSURII  
- Europe (Ireland)  
  arn:aws:controltower:eu-west-1::control/SGDBXEOLRRUY  
- Europe (London)  
  arn:aws:controltower:eu-west-2::control/BICLJDRGFKNI  
- Europe (Stockholm)  
  arn:aws:controltower:eu-north-1::control/GKRGBFPIQRTMC  
- Asia Pacific (Mumbai)  
  arn:aws:controltower:ap-south-1::control/IQOKDDFQVULP  
- Asia Pacific (Seoul)  
  arn:aws:controltower:ap-northeast-2::control/HANQYUPDQHSI  
- Asia Pacific (Tokyo)  
  arn:aws:controltower:ap-northeast-1::control/OWTLNREXRIMN  
- Europe (Paris)  
  arn:aws:controltower:eu-west-3::control/QASOHDPDRFTMQ  
- South America (São Paulo)  
  arn:aws:controltower:sa-east-1::control/REZTDFNCSTEQQ  
- US West (N. California) |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Hong Kong)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/IRNOVPOTKTKX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Jakarta)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/KVIRUTURJNHT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Osaka)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/MYGXAKQZZKAU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Europe (Milan)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/IOUXAVZFJDAK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Africa (Cape Town)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/UWOYYBYLAJQS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Middle East (Bahrain)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/RMEBEBQJNGW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Israel (Tel Aviv)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/TYGLPCXHNFSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Europe (Zurich)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/ADCWXSRS4WJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Europe (Spain)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/DICUTESJCRB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/RIKHLWLBSQOP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Middle East (UAE)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/CSXMSFJQHGCV</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>CT.KMS.PR.3</td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>southeast-4::control/VIADYMCEWUAQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>east-1::control/SILIQXPGEKMKZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>east-2::control/GNVSURRRJVKK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/MXVGDDITBSPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/OGEDACIRUOBW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/GCUVYVGGBEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/XTNLHQDKVOZW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/UYOKYCLBZTQR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/KSSFTGALEFZC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/FSWTVZUNYVSF</td>
</tr>
</tbody>
</table>

CT.KMS.PR.3

- NIST 800-53 Rev 5 AC-2
- NIST 800-53 Rev 5 AC-2(1)
- NIST 800-53 Rev 5 AC-3
- NIST 800-53 Rev 5 AC-3(15)
- NIST 800-53 Rev 5 AC-3(7)
- NIST 800-53 Rev 5 AC-5
- NIST 800-53 Rev 5 AC-6
- NIST 800-53 Rev 5 AC-6(3)
- PCI DSS version 3.2.1 2.2
- PCI DSS version 3.2.1 7.1.1
- PCI DSS version 3.2.1 7.1.2
- PCI DSS version 3.2.1 7.2.1
- PCI DSS version 3.2.1 7.2.2

- Enforce least privilege

1930
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/FWPBETJPERDF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/VEZBRKXRLKAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/SDHRAOXPCXDR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/WQMVYTCUAPHO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/NXGJVOLKXVTZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/YYGKNXYYZMUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/VJFECXNNFSLK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/CTNRPSCIMTII</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/HXBWEHKKWQBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/CDANCVXVACTRP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/QLCVBYTAQKNZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.LAMBDA.PR.2</td>
<td></td>
<td>- Limit network access</td>
<td>- US East (N. Virginia) arn:aws:controltower:us-east-1::control/GFCPWDHPFTFD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- US East (Ohio) arn:aws:controltower:us-east-2::control/COUMHGZRQFCF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- US West (Oregon) arn:aws:controltower:us-west-2::control/GDTEOOGPCW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/BOMYPAZQNZU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Middle East (UAE) arn:aws:controltower:me-central-1::control/WHNUPURPZJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/ZFKVOWDNSYFB</td>
</tr>
<tr>
<td>CT.LAMBDA.PR.2</td>
<td>NIST 800-53 Rev 5 AC-21</td>
<td>- US East (N. Virginia) arn:aws:controltower:us-east-1::control/GFCPWDHPFTFD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>- US East (Ohio) arn:aws:controltower:us-east-2::control/COUMHGZRQFCF</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>- US West (Oregon) arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>- Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/BOMYPAZQNZU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Middle East (UAE) arn:aws:controltower:me-central-1::control/WHNUPURPZJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/ZFKVOWDNSYFB</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td>West-2::control/ VJIRQWYHYDXV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• NIST 800-53 Rev 5
• PCI DSS version 3.2.1
• PCI DSS version 3.2.1
• PCI DSS version 3.2.1
• PCI DSS version 3.2.1
• PCI DSS version 3.2.1
• PCI DSS version 3.2.1
• PCI DSS version 3.2.1
• PCI DSS version 3.2.1
• PCI DSS version 3.2.1
• PCI DSS version 3.2.1

• Canada (Central)
  arn:aws:controltower:ca-central-1::control/
  FXFPBFXFQPQWW

• Asia Pacific (Sydney)
  arn:aws:controltower:ap-southeast-2::control/
  BAGTSQJJEERJ

• Asia Pacific (Singapore)
  arn:aws:controltower:ap-southeast-1::control/
  LDUXXPHTFHTM

• Europe (Frankfurt)
  arn:aws:controltower:eu-central-1::control/
  HLIQPFVKCLVE

• Europe (Ireland)
  arn:aws:controltower:eu-west-1::control/
  MJBFVBCQCLFLW

• Asia Pacific (Mumbai)
  arn:aws:controltower:ap-south-1::control/
  YMVFZTVHNPMMQ

• Asia Pacific (Seoul)
  arn:aws:controltower:ap-northeast-2::control/
  LNBQPFQPQISPJ

• Asia Pacific (Tokyo)
  arn:aws:controltower:ap-northeast-1::control/
  PUXILAKLQQYU

• Europe (Paris)
  arn:aws:controltower:eu-west-3::control/
  IRVLXVEJRTUD

• South America (São Paulo)
  arn:aws:controltower:sao-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>MKVUCRWVYEWG</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td><strong>US West (N. California)</strong></td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/YSIDNPMMYOON</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Hong Kong)</strong></td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/JVMXSVKDKFOU</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Jakarta)</strong></td>
</tr>
<tr>
<td>(Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/UXZDCQJYOSNC</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Osaka)</strong></td>
</tr>
<tr>
<td>(Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/VYEPJCYALYAK</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td><strong>Europe (Milan)</strong></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-1::control/BORKYWUAXJKK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape</td>
<td></td>
<td></td>
<td><strong>Africa (Cape Town)</strong></td>
</tr>
<tr>
<td>Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/UIBDVHMMVAUF</td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td></td>
<td><strong>Middle East (Bahrain)</strong></td>
</tr>
<tr>
<td>(Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/RMYYTDDBRHYMM</td>
</tr>
<tr>
<td>• Israel (Tel</td>
<td></td>
<td></td>
<td><strong>Israel (Tel Aviv)</strong></td>
</tr>
<tr>
<td>Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/XJMPXBDGZCY</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td><strong>Europe (Zurich)</strong></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-central-2::control/XLAKVGWNSZVU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td><strong>Europe (Spain)</strong></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-2::control/LPJIPJVEVHOK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
</tr>
<tr>
<td>(Hyderabad)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/XWZGMYTSZAC</td>
</tr>
</tbody>
</table>
## CT.LAMBDA.PR.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/HBVCDXCMWQXG</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/AJDRYJFPHMSW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/OYLHPRXKQADB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/WHXWCLBQBWBC</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/WQQSTHULHXBB</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/IJVOXESXUCQJ</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/ANRAIFWUPJNN</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/GPNJQIYPIURY</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/PZWTIELYZTE</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/CLTIPWNNUWFU</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/MWERAVUHAGVJ</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>Europe (Milan) arn:aws:controltower:eu-west-1::control/GPNJQIYPIURY</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.LAMBDA.PR.4</td>
<td></td>
<td></td>
<td>south-1::control/OUQCUGMLXQY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/QRUBHQOLDEO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/ZQNEWQONHOCA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/XVIMEZMIVPD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eucentral-2::control/WKGMNYMTIOUY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eusouth-2::control/YVGINZGKBDRB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:apsouth-2::control/OUAKOUKARUHA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/MSMKHOTQFYUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:apsoutheast-4::control/RPXKXCYTLEA</td>
</tr>
</tbody>
</table>

**CT.LAMBDA.PR.4**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.LAMBDA.PR.4</td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 AC-21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enforce least privilege</td>
<td>• US East (N. Virginia)   arn:aws:controltower:useast-1::control/XOXGPEWXYQ</td>
</tr>
</tbody>
</table>
|                    |           |                   | • US East (Ohio)                arn:aws:controltower:us-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>east-2::control/RNARZBOWKHIP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/PAVLGMUCWQJJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/PMHQAFGLASNE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/DFIQWZHSSNZY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/JRQNMMQUXFVX</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:europe-central-1::control/PACONHNZQYAF</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:europe-west-1::control/MYNQPUTAPPHV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:europe-west-2::control/QLUVOLECEOBI</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:europe-north-1::control/IIPLOXPQIWIC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/BZXKMLBAUXJY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/JFSZBXDKTWKB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/UKOEFDBEQKA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:europe-north-1::control/UKOEFDBEQKA</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>west-3::control/</td>
<td></td>
<td></td>
<td>UUIQOLJJWCSO</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/ QFUSEDARBJFK</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/ OQKNXSQRWEEO</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/ GMZAGJHNXSKP</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/ ZGTSAFJLTDNER</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/ PJEENJYYVITLJK</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/ YYQIXBRIPJHJ</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/ JREQFLRKLPTLJ</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/ ZRACMNDXAWNY</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/ UMCFTGGINALH</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eucentral-2::control/ WWDZVOAOVTRP</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eusouth-2::control/ DTWVWLVUTWKG</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.LAMBDA.PR.5</td>
<td>NIST 800-53 Rev 5 AC-2(1)</td>
<td>Enforce least privilege</td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Middle East (UAE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Melbourne)</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/ MDQQQEWXASLO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/ YLQGBZVFJHNF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/ SYNEUHQJIFI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/ PCXKSQRFQJG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/ YLLDFQWUTMQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/ BEAYVEJMKAZH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/ GJHAEAJFHIIV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/ DNBATCDTCWHT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/ OFUJNIZKUUTH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/ TOUJWQCMHRT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/ PMHEGULRZVPO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.LAMBDA.PR.6</td>
<td>NIST 800-53 Rev 5 AC-21</td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>east-1::control/</td>
</tr>
<tr>
<td>AC-3</td>
<td></td>
<td></td>
<td>UZMMJBQMQZK</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td>AC-3(7)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>east-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KOWWBEWWDHEQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td>AC-4</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>west-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SRWDRZQEYMNCW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td>SC-7</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DIYTGMAGAPPVG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td>SC-7(11)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>southeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DOXCJUNEAQCI</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>SC-7(16)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>southeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VITKCEWDMFWW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td>SC-7(20)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DEFEEGBBWNMG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td>SC-7(21)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>QXXQDGAQAXXW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td>SC-7(3)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>west-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JZTICQHAKTAT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td>SC-7(4)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>north-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CHNMHTENMIPB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>1.2.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>XORGKKZRAMGT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>1.3</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>northeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NFIDIZDUGHITE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>1.3.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>northeas-1::control/KGDGFZTQXAD</td>
<td>• Europe (Paris)</td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/BGUYDODQXMMF</td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo)</td>
<td></td>
<td>arn:aws:controltower:saeast-1::control/NECDSRJUJGJP</td>
</tr>
<tr>
<td></td>
<td>• US West (N. California)</td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/OLJVZHFBBQX</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/DMNSQNZNIVC</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/LVDYJRBNZIDJ</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/QIZTBZNUZBXX</td>
</tr>
<tr>
<td></td>
<td>• Europe (Milan)</td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/QAMSLUZWJHTS</td>
</tr>
<tr>
<td></td>
<td>• Africa (Cape Town)</td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/DEWAWECDWNWD</td>
</tr>
<tr>
<td></td>
<td>• Middle East (Bahrain)</td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/KHSXAHNQPIF</td>
</tr>
<tr>
<td></td>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/FS0FPVGVUXHI</td>
</tr>
<tr>
<td></td>
<td>• Europe (Zurich)</td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/WIIRwsRTEYBB</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.NEPTUNE.PR.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td>AC-2(1)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VVYTTDIRYZIC</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td>AC-3(15)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>east-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HHYQNNQRWPTR</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td>AC-3(7)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>west-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OIAIWBDRZL</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td>AC-6</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IMAPSYUKKCBZ</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>southeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ZBNZCKKEFZPZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>southeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UKYLCMU2FQMF</td>
</tr>
<tr>
<td>• Enforce least</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>privilege</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Use strong</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>authentication</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/TOIBCMWTQULC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/QBMPFYNYANKO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/TQGVMTLQCLVJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/JCSHAIJVQJP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/HKUTEITUOURQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/FIHJPPXZCWUDV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/WMPRSCMJQPR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/XPUCIKNRWZQY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/BIMAERWOCWOC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/AGSWUBESEOXXW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/BQEHCGQVUZAL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Asia Pacific (Jakarta) | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| southeast-3::control/ZCYILJXNYJPS | • Asia Pacific (Osaka)  
  arn:aws:controltower:ap-northeast-3::control/VRVLPEXHUHRR | | |
| | • Europe (Milan)  
  arn:aws:controltower:eu-south-1::control/VSKCNZBJMSDL | | |
| | • Africa (Cape Town)  
  arn:aws:controltower:af-south-1::control/ERYIKQHLG0SR | | |
| | • Middle East  
  (Bahrain)  
  arn:aws:controltower:me-south-1::control/JAEBJWGZUZIK | | |
| | • Israel (Tel Aviv)  
  arn:aws:controltower:il-central-1::control/XNEQNATMDCOH | | |
| | • Europe (Zurich)  
  arn:aws:controltower:eu-central-2::control/XYVGUQMPBHL0 | | |
| | • Europe (Spain)  
  arn:aws:controltower:eu-south-2::control/EDCFDZFLSLTE | | |
| | • Asia Pacific  
  (Hyderabad)  
  arn:aws:controltower:ap-south-2::control/GCIDLPWUZZSC | | |
| | • Middle East (UAE)  
  arn:aws:controltower:me-central-1::control/DMMWWJNLMZZA | | |
| | • Asia Pacific  
  (Melbourne)  
  arn:aws:controltower:ap-southeast-4::control/EMRQQNEEYXAN | | |
### Control identifier

**CT.NEPTUNE.PR.2**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.NEPTUNE.PR.2    | • NIST 800-53 Rev 5 CA-9(1)  
                   |           | • Improve availability  
                   |                     | • Protect configurations  |
|                    | • NIST 800-53 Rev 5 CM-2  
                   |           |                     | • US East (N. Virginia)  
                   |                     | arn:aws:controltower:us-east-1::control/AULKJ0KXSVDI  
                   |                     | • US East (Ohio)  
                   |                     | arn:aws:controltower:us-east-2::control/ZHXCJVNWNCCQ  
                   |                     | • US West (Oregon)  
                   |                     | arn:aws:controltower:us-west-2::control/GPMNZXUWJXSR  
                   |                     | • Canada (Central)  
                   |                     | arn:aws:controltower:ca-central-1::control/VFQGQVHFGGE  
                   |                     | • Asia Pacific (Sydney)  
                   |                     | arn:aws:controltower:ap-southeast-2::control/EPHJMXQEBVJC  
                   |                     | • Asia Pacific (Singapore)  
                   |                     | arn:aws:controltower:ap-southeast-1::control/MKEKOUWXVHEC  
                   |                     | • Europe (Frankfurt)  
                   |                     | arn:aws:controltower:eu-central-1::control/BF01LRCAVCCO  
                   |                     | • Europe (Ireland)  
                   |                     | arn:aws:controltower:eu-west-1::control/WXRTDNJWPQOX  
                   |                     | • Europe (London)  
                   |                     | arn:aws:controltower:eu-west-2::control/OCSPXMPOABNK  
                   |                     | • Europe (Stockholm)  
                   |                     | arn:aws:controltower:eu-north-1::control/LZEHPWXYBNV  
                   |                     | • Asia Pacific (Mumbai)  
                   |                     | arn:aws:controltower:ap-south-1::control/USKYFIQHPSRG  |
|                    | • NIST 800-53 Rev 5 CM-2(2)  
                   |           |                     | • US East (N. Virginia)  
                   |                     | arn:aws:controltower:us-east-1::control/AULKJ0KXSVDI  
                   |                     | • US East (Ohio)  
                   |                     | arn:aws:controltower:us-east-2::control/ZHXCJVNWNCCQ  
                   |                     | • US West (Oregon)  
                   |                     | arn:aws:controltower:us-west-2::control/GPMNZXUWJXSR  
                   |                     | • Canada (Central)  
                   |                     | arn:aws:controltower:ca-central-1::control/VFQGQVHFGGE  
                   |                     | • Asia Pacific (Sydney)  
                   |                     | arn:aws:controltower:ap-southeast-2::control/EPHJMXQEBVJC  
                   |                     | • Asia Pacific (Singapore)  
                   |                     | arn:aws:controltower:ap-southeast-1::control/MKEKOUWXVHEC  
                   |                     | • Europe (Frankfurt)  
                   |                     | arn:aws:controltower:eu-central-1::control/BF01LRCAVCCO  
                   |                     | • Europe (Ireland)  
                   |                     | arn:aws:controltower:eu-west-1::control/WXRTDNJWPQOX  
                   |                     | • Europe (London)  
                   |                     | arn:aws:controltower:eu-west-2::control/OCSPXMPOABNK  
                   |                     | • Europe (Stockholm)  
                   |                     | arn:aws:controltower:eu-north-1::control/LZEHPWXYBNV  
                   |                     | • Asia Pacific (Mumbai)  
                   |                     | arn:aws:controltower:ap-south-1::control/USKYFIQHPSRG  |
|                    | • NIST 800-53 Rev 5 CM-3  
                   |           |                     | • US East (N. Virginia)  
                   |                     | arn:aws:controltower:us-east-1::control/AULKJ0KXSVDI  
                   |                     | • US East (Ohio)  
                   |                     | arn:aws:controltower:us-east-2::control/ZHXCJVNWNCCQ  
                   |                     | • US West (Oregon)  
                   |                     | arn:aws:controltower:us-west-2::control/GPMNZXUWJXSR  
                   |                     | • Canada (Central)  
                   |                     | arn:aws:controltower:ca-central-1::control/VFQGQVHFGGE  
                   |                     | • Asia Pacific (Sydney)  
                   |                     | arn:aws:controltower:ap-southeast-2::control/EPHJMXQEBVJC  
                   |                     | • Asia Pacific (Singapore)  
                   |                     | arn:aws:controltower:ap-southeast-1::control/MKEKOUWXVHEC  
                   |                     | • Europe (Frankfurt)  
                   |                     | arn:aws:controltower:eu-central-1::control/BF01LRCAVCCO  
                   |                     | • Europe (Ireland)  
                   |                     | arn:aws:controltower:eu-west-1::control/WXRTDNJWPQOX  
                   |                     | • Europe (London)  
                   |                     | arn:aws:controltower:eu-west-2::control/OCSPXMPOABNK  
                   |                     | • Europe (Stockholm)  
                   |                     | arn:aws:controltower:eu-north-1::control/LZEHPWXYBNV  
                   |                     | • Asia Pacific (Mumbai)  
                   |                     | arn:aws:controltower:ap-south-1::control/USKYFIQHPSRG  |
|                    | • NIST 800-53 Rev 5 SC-5(2)  
                   |           |                     | • US East (N. Virginia)  
                   |                     | arn:aws:controltower:us-east-1::control/AULKJ0KXSVDI  
                   |                     | • US East (Ohio)  
                   |                     | arn:aws:controltower:us-east-2::control/ZHXCJVNWNCCQ  
                   |                     | • US West (Oregon)  
                   |                     | arn:aws:controltower:us-west-2::control/GPMNZXUWJXSR  
                   |                     | • Canada (Central)  
                   |                     | arn:aws:controltower:ca-central-1::control/VFQGQVHFGGE  
                   |                     | • Asia Pacific (Sydney)  
                   |                     | arn:aws:controltower:ap-southeast-2::control/EPHJMXQEBVJC  
                   |                     | • Asia Pacific (Singapore)  
                   |                     | arn:aws:controltower:ap-southeast-1::control/MKEKOUWXVHEC  
                   |                     | • Europe (Frankfurt)  
                   |                     | arn:aws:controltower:eu-central-1::control/BF01LRCAVCCO  
                   |                     | • Europe (Ireland)  
                   |                     | arn:aws:controltower:eu-west-1::control/WXRTDNJWPQOX  
                   |                     | • Europe (London)  
                   |                     | arn:aws:controltower:eu-west-2::control/OCSPXMPOABNK  
                   |                     | • Europe (Stockholm)  
                   |                     | arn:aws:controltower:eu-north-1::control/LZEHPWXYBNV  
                   |                     | • Asia Pacific (Mumbai)  
                   |                     | arn:aws:controltower:ap-south-1::control/USKYFIQHPSRG  |
|                    | • PCI DSS version 3.2.1  
                   |           |                     | • US East (N. Virginia)  
                   |                     | arn:aws:controltower:us-east-1::control/AULKJ0KXSVDI  
                   |                     | • US East (Ohio)  
                   |                     | arn:aws:controltower:us-east-2::control/ZHXCJVNWNCCQ  
                   |                     | • US West (Oregon)  
                   |                     | arn:aws:controltower:us-west-2::control/GPMNZXUWJXSR  
                   |                     | • Canada (Central)  
                   |                     | arn:aws:controltower:ca-central-1::control/VFQGQVHFGGE  
                   |                     | • Asia Pacific (Sydney)  
                   |                     | arn:aws:controltower:ap-southeast-2::control/EPHJMXQEBVJC  
                   |                     | • Asia Pacific (Singapore)  
                   |                     | arn:aws:controltower:ap-southeast-1::control/MKEKOUWXVHEC  
                   |                     | • Europe (Frankfurt)  
                   |                     | arn:aws:controltower:eu-central-1::control/BF01LRCAVCCO  
                   |                     | • Europe (Ireland)  
                   |                     | arn:aws:controltower:eu-west-1::control/WXRTDNJWPQOX  
                   |                     | • Europe (London)  
                   |                     | arn:aws:controltower:eu-west-2::control/OCSPXMPOABNK  
                   |                     | • Europe (Stockholm)  
                   |                     | arn:aws:controltower:eu-north-1::control/LZEHPWXYBNV  
                   |                     | • Asia Pacific (Mumbai)  
                   |                     | arn:aws:controltower:ap-south-1::control/USKYFIQHPSRG  |

---

**1948**
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ WBFRRUHLHDLQY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ NACPTUROMGYT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ ILBYOEHNJWP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ VHPBMTMAGERS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ YSOAFXOUVDMZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ MNXLYFLERVOH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ MQCMSWZMSNP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ RBZJYTJKEHN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/ CPEMPXZLTSTB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/ NCHILLHMRKwV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/ CITQXSOCSZKP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CT.NEPTUNE.PR.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.NEPTUNE.PR.3</td>
<td></td>
<td>Encrypt data at rest</td>
<td>Central-1::control/TYXJKFHWUHHA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eucentral-2::control/JP0GMCFZRQDK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eusouth-2::control/JHJZOWOQKSQQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:apsouth-2::control/ALDJGKDLKYSY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/PAKICJQWADVT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:apseast-4::control/IJPZWNXPUXES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (N. Virginia) arn:aws:controltower:useast-1::control/ZVGXWMKLNOCT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:useast-2::control/DUNNSUBEGAMH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:useast-2::control/CKBZYEFYGFQG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:cenentral-1::control/BONSNFHFTQYM</td>
</tr>
</tbody>
</table>
|                   |           |                   | • Asia Pacific (Sydney) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>southeast-2::control/ GQZRECITMLZC</td>
<td>• Asia Pacific (Singapore)</td>
<td>turnaround::control/ HXRJRJSRDQBB</td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ GQXVHTLKFUL0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ GFECITKOJIDB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ VRONLZOJVXSF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ RGGBNARHYLPC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ DSCGRLDBSTVB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ MPXCNSRKMDFR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ LPFIVZUR0OJJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/ EEF0BFRQOZLO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/ PHRBYWIDRHTW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/ CZOFYPWHMMZD</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/THFQXQZMWHWZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/WUVFSNJYPXZY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/MEYIGJFZNYVR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/VELCDMBUYSDH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/MAJYUEOFRPCV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/POYFLOMZJKGT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/TGFIVIAQMYYH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/DATHNOLBMKAS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/UMJFXSSLQXOZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/AUXNHIKWyW0M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/XVYPPSSGWQEMK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Melbourne)</td>
<td>arn:aws:controltower:ap-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
CT.NEPTUNE.PR.4 | • NIST 800-53 Rev 5 AC-2(4)  
• NIST 800-53 Rev 5 AC-4(26)  
• NIST 800-53 Rev 5 AC-6(9)  
• NIST 800-53 Rev 5 AU-10  
• NIST 800-53 Rev 5 AU-12  
• NIST 800-53 Rev 5 AU-2  
• NIST 800-53 Rev 5 AU-3  
• NIST 800-53 Rev 5 AU-6(1)  
• NIST 800-53 Rev 5 AU-6(3)  
• NIST 800-53 Rev 5 AU-6(4)  
• NIST 800-53 Rev 5 AU-6(5)  
• NIST 800-53 Rev 5 AU-7(1)  
• NIST 800-53 Rev 5 AU-9(7)  
• NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 SC-7(9)  
• NIST 800-53 Rev 5 SI-20  
• NIST 800-53 Rev 5 SI-3(8)  
• NIST 800-53 Rev 5 SI-4(20)  
• NIST 800-53 Rev 5 SI-4(5)  
• Establish logging and monitoring | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/ONFZFWFIMSQN  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/JDYGXWIKJYDZ  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/JSFACLSPAAMK  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/ONFZFWFIMSQN  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/ICNUQORDNDTA  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/WNMAYPTTNDEJ  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/ZLNXPZLYQ0KG  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/XWJ1DBFMFSIU  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/ROEMRZG0KQFG  
• Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/CQKTUQPFGNZ.
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controlledtower:ap-south-1::control/EKPWQ3MYXYJC</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controlledtower:ap-northeast-2::control/GJEVNGPGKXIW</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controlledtower:ap-northeast-1::control/TTUQITYQNLQX</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.2.2</td>
<td></td>
<td>• Europe (Paris) arn:aws:controlledtower:eu-west-3::control/KMOMZHHLSJWK</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td></td>
<td>• South America (São Paulo) arn:aws:controlledtower:sa-east-1::control/LAOWKRUXEWCI</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td>• US West (N. California) arn:aws:controlledtower:us-west-1::control/ZAHCCDVLHWGD</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controlledtower:ap-east-1::control/CVRZOMASWHHH</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.2.7</td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controlledtower:ap-southeast-3::control/IOWUKZDWGONS</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controlledtower:ap-northeast-3::control/CETXVRVQUKY</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td></td>
<td>• Europe (Milan) arn:aws:controlledtower:eu-south-1::control/OLIVTOIMRZCM</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controlledtower:af-south-1::control/QXMAAXPCKZFS</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controlledtower:me-</td>
</tr>
</tbody>
</table>
CT.NEPTUNE.PR.5

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.NEPTUNE.PR.5</td>
<td></td>
<td>Improve resiliency</td>
<td></td>
</tr>
</tbody>
</table>

- NIST 800-53 Rev 5 SI-12
- PCI DSS version 3.2.1 3.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>south-1::control/ PGRATNEIEHKBS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/ZHQJPXBOEXCR</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/XKPKBPMMDZKQ</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/JOVZFQYFBTRR</td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/SQJLLQWWDYLT</td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/PHQRDGGBMVYVR</td>
</tr>
<tr>
<td>• Asia Pacific (Melbourne)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/EKFELZAAQZPG</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>central-1::control/VRSCWUWXSK00</td>
<td>• Asia Pacific (Sydney)</td>
<td>• Asia Pacific (Singapore)</td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td>VRSCWUWXSK00</td>
<td>arn:aws:controltower:ap-southeast-2::control/HNYWAXKKADCC</td>
<td>arn:aws:controltower:ap-southeast-1::control/LFORACIZCHRB</td>
<td>arn:aws:controltower:eucentral-1::control/HJZPJJVXQESR</td>
</tr>
<tr>
<td>arn:aws:controltower:eu-west-1::control/DGTIMIBUIKZV</td>
<td>• Europe (Ireland)</td>
<td>• Europe (London)</td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>• Asia Pacific (Seoul)</td>
<td>• Asia Pacific (Tokyo)</td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>• US West (N. California)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KAOCDXVZVGYR</td>
<td>arn:aws:controltower:sa-east-1::control/KAOCDXVZVGYR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/IXTSMKZWHMG</td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/VGEOCPJNBAUE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/OTVNXEOUUPBR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/AFEXSGMDSATC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/KAONKKJAVXIIY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/RBOCORVIYGTI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/FHDEQVJNEKKO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/UEFOVDHPYOP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/NEJJWTJPTYO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/EIEQGAVIHYZT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/GUZDYGGUCQEF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/LVFHIRHHXDNP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.NETWORK-FIREWALL.PR.1 | • NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-2  
• PCI DSS version 3.2.1 1.3.6  
• PCI DSS version 3.2.1 11.4 | • Limit network access |  
• US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/ MOWFCDWVPVGY  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/ GCCHBPSTREXX  
• US West (Oregon)  
arn:aws:aws:controltower:us-west-2::control/ AHRCA BRNZNQM  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/ SIIOUKQWZSFK  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/ ITJWVPYITYEB  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/ VALVPGFBFRBG  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/ JOWPYQPSKHFY  
• Europe (Ireland)  
arn:aws:aws:controltower:eu-west-1::control/ JCRWUNVVL OXY  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/ ISNZYDYDKGKX |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ HUIWIDBGYYVJB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ SEWCMVQOWXZF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ VTAPSFMTMEPO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ MKTMBJSACTFI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ VOSZJFYQWKNN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ XAHWAUUTZFGI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ RSWJSOQSPPLGG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ CEMWKHOGONEX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ ETUWZLERTVQI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ KFPGHTQXQZEY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/ HOIIEWDTLWCMZ</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Africa (Cape Town) | arn:aws:controltower:af-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.NETWORK-FIREWALL.PR.2 | • NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-2  
• PCI DSS version 3.2.1 1.2.1  
• PCI DSS version 3.2.1 1.3 | • Limit network access | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
SXRZLKJBMWDWS  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
IWTSV0SLDTBS  
• US West (Oregon)  
arn:aws:controltower:us-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>west-2:control/DBQINFJQURT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>Canada (Central)arn:aws:controltower:ca-central-1:control/GTFHCKELKRXJ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)arn:aws:controltower:ap-southeast-2:control/JTCRHWTVYYMB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)arn:aws:controltower:ap-southeast-1:control/IZHFDRGPH3YF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt)arn:aws:controltower:eu-central-1:control/GTUXAMBREPYK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland)arn:aws:controltower:eu-west-1:control/HOVYKEEFT1GU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London)arn:aws:controltower:eu-west-2:control/KDGIPSOLEUPZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm)arn:aws:controltower:eu-north-1:control/IZLEXYHKBEBBOW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)arn:aws:controltower:ap-south-1:control/ZJXSBKPKNOO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)arn:aws:controltower:ap-northeast-2:control/XXDJDIPDXVVT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo)arn:aws:controltower:ap-northeast-1:control/SDCRNKFTGMD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris)arn:aws:controltower:eu-west-3:control/TMFXXPMElCKA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo)arn:aws:controltower:sa-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>east-1::control/ERXMHKYKAWSC</td>
<td><strong>US West (N. California)</strong></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/NKALWPJQEMGP</td>
</tr>
<tr>
<td></td>
<td><strong>Asia Pacific (Hong Kong)</strong></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/IGGMCFFBBHYCG</td>
</tr>
<tr>
<td></td>
<td><strong>Asia Pacific (Jakarta)</strong></td>
<td></td>
<td>arn:aws:aws:controltower:ap-southeast-3::control/MTPRETWPQRQNM</td>
</tr>
<tr>
<td></td>
<td><strong>Asia Pacific (Osaka)</strong></td>
<td></td>
<td>arn:aws:aws:controltower:ap-northeast-3::control/VUNCRJLJDFIO</td>
</tr>
<tr>
<td></td>
<td><strong>Europe (Milan)</strong></td>
<td></td>
<td>arn:aws:aws:controltower:eusouth-1::control/ERYDUNHYMURH</td>
</tr>
<tr>
<td></td>
<td><strong>Africa (Cape Town)</strong></td>
<td></td>
<td>arn:aws:aws:controltower:af-south-1::control/RSVJLVHRYQNL</td>
</tr>
<tr>
<td></td>
<td><strong>Middle East (Bahrain)</strong></td>
<td></td>
<td>arn:aws:aws:controltower:messouth-1::control/FOETEMT2EZRFR</td>
</tr>
<tr>
<td></td>
<td><strong>Israel (Tel Aviv)</strong></td>
<td></td>
<td>arn:aws:aws:controltower:il-central-1::control/TKQ2NLWJYRZK</td>
</tr>
<tr>
<td></td>
<td><strong>Europe (Zurich)</strong></td>
<td></td>
<td>arn:aws:aws:controltower:eucentral-2::control/ZLWUEANEUGKI</td>
</tr>
<tr>
<td></td>
<td><strong>Europe (Spain)</strong></td>
<td></td>
<td>arn:aws:aws:controltower:eusouth-2::control/BUMCKVFUMSZX</td>
</tr>
<tr>
<td></td>
<td><strong>Asia Pacific (Hyderabad)</strong></td>
<td></td>
<td>arn:aws:aws:controltower:ap south-2::control/HYPTJDAYXNNH</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.NETWORK-FIREWALL.PR.3 | • NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-2  
• PCI DSS version 3.2.1 1.2.1  
• PCI DSS version 3.2.1 1.3  
• PCI DSS version 3.2.1 1.3.1  
• PCI DSS version 3.2.1 1.3.2  
• PCI DSS version 3.2.1 1.3.4  
• PCI DSS version 3.2.1 1.3.6 | • Limit network access | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/POSYWADLSXI  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/QCOLRTHQOZKD  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/CBTUDPFAKVKC  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/YZYEDZMRQLKG  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/JFLLDLXJUPDD  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/TKVYXDHYRAET  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/EUXJZRQMLYNG  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/WQSZOX0JGABB |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/RLWYKMF0MTHK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/IXIEKMH0PXA</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/VBXMAKCXQCPJ</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/NPXRXMOWRHB</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/XFESNQJWYRZV</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/LAGHSXBMNPCR</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/HGQP0DHBJRUD</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/HWHGLJZTLCGZ</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/CWCHP0D01PCBB</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/ISPNAEMQOYUG</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/VCMMKLXPDXB5</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/1964</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.NETWORK-FIREWALL.PR.4</td>
<td>NIST 800-53 Rev 5 AC-4(21)</td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/OFYAEXY5CB0V</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-1::control/OFYAEXY5CB0V</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.NETWORK-FIREWALL.PR.4</td>
<td>NIST 800-53 Rev 5 AC-4(21)</td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/OFYAEXY5CB0V</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-1::control/OFYAEXY5CB0V</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td>east-2::control/KNIMX0HJDVNW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/VOEDZGRSHEV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/QB0RBGFWF9IX</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(5)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/B02PZUYFXHUQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AFHBFGRQFBMV</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/TYHKWQJMOLGM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/GPNDNENLPWWN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/QCBVHDIASDKR</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/CS1PNPQGGXWV</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/VZAFYQVTASYJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/NVRIPMNUNUX5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ETWPTTQYSUZG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:europe-central-1::control/ARWIDSNRQX78</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/QB0RBGFWF9IX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/B02PZUYFXHUQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AFHBFGRQFBMV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/TYHKWQJMOLGM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/GPNDNENLPWWN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/QCBVHDIASDKR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/CS1PNPQGGXWV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/VZAFYQVTASYJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/NVRIPMNUNUX5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ETWPTTQYSUZG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:europe-central-1::control/ARWIDSNRQX78</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>west-3::control/PSXIXHKFMZYD</td>
<td>• South America (São Paulo)</td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/TQEZIBLEAPLP</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/CWUANEFSYHT</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/OXBWNYZSDNXJ</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/PGEABSLQWWSZ</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/TXSZCXGZSSN</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/VFYJRIPXCTJ</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/QPYERNZE0VFW</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/VAMNEBFICAGE</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/ITFJURZDNXBC</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/HYYFUKRTDISL</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/YTUMVCGZMBR</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.1</td>
<td>• NIST 800-53 Rev 5 AC-17(2)</td>
<td>• Encrypt data at rest</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/RIIBJUAGRIHM</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ZKJKSTRBBZVZ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-2(4)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/LIWOVOJLBAX</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-21</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/DYHUHTZMANYT</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ZKCLZQKAZSTAI</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/VIMTCPWLJSNN</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/KYNVRIQMLTBN</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(26)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6(9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/MUMEKFGKATWE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/RYGXUZLZWL5W</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/LFGRRNFRJMKXE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ZBNHDIO26ZEMY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/FGYTCVPCXE6E</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/XFOEOJKEETKE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/QUXXZOD0NXRY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/PHABICPTLMTC</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-10</td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/DZKPTXADQ8LKL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/TNQHFJXDBPS</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 IA-5(1)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/BZMOURQRWBKA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-12(3)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-jp::control/</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8</td>
<td>• NIST 800-53 Rev 5 SC-8(1)</td>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td>• PCI DSS version 3.2.1 1.3</td>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td>• PCI DSS version 3.2.1 1.3.2</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td>• PCI DSS version 3.2.1 10.1</td>
<td>• PCI DSS version 3.2.1 10.2.1</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.2</td>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td>• PCI DSS version 3.2.1 10.2.5</td>
</tr>
</tbody>
</table>

- **northeast-3::control/TEXJPFYIYIRT**
- **Europe (Milan) arn:aws:controltower:eu-south-1::control/BKTRNYRJZ2C**
- **Africa (Cape Town) arn:aws:controltower:af-south-1::control/SQGYKIPV6STL**
- **Middle East (Bahrain) arn:aws:controltower:me-south-1::control/HGHTOHZXLQHR**
- **Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/OSIPIFATXXH0**
- **Europe (Zurich) arn:aws:controltower:eu-central-2::control/YKSFVHHVRUZ**
- **Europe (Spain) arn:aws:controltower:eu-south-2::control/RQXYFZTDBGCS**
- **Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/STQLABAVRXQO**
- **Middle East (UAE) arn:aws:controltower:me-central-1::control/TBXMTAFVCDWJ**
- **Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/QGIEHLWASGGX**
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.OPENSEARCH.PR.10 | • NIST 800-53 Rev 5 AC-21<br> • NIST 800-53 Rev 5 AC-3<br> • NIST 800-53 Rev 5 AC-3(7)<br> • NIST 800-53 Rev 5 AC-4<br> • NIST 800-53 Rev 5 AC-4(21)<br> • NIST 800-53 Rev 5 AC-6 | • Limit network access | • US East (N. Virginia)<br>  arn:aws:controltower:us-east-1::control/AASJRFPGBVMO<br> • US East (Ohio)<br>  arn:aws:controltower:us-east-2::control/QZXAALUVUXXP<br> • US West (Oregon)<br>  arn:aws:controltower:us-west-2::control/SMJAAEIMFEDL<br> • Canada (Central)<br>  arn:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>central-1::control/CKNPGFNVIVZ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/JUUIFSJWJWIT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/LLKQQRPPWTV</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/CYKUSFNSTJZZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/TJZKDXZQKHAF</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/YVKPKMDZGQIIW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/FOCMHCPHMGIA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/VOXPJBUXNXSH</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/XRNYJUFDPRET</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/YTWEMPALPIMN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/KQIMWODUHUH</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/UCCGQWSINGEF</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
|                    |           |                  | *Asia Pacific (Hong Kong)*
|                    |           |                  | `arn:aws:controltower:ap-east-1::control/EKBQJYXUEQMK` |
|                    |           |                  | *Asia Pacific (Jakarta)*
|                    |           |                  | `arn:aws:controltower:ap-southeast-3::control/WTKHFUNAHSRC` |
|                    |           |                  | *Asia Pacific (Osaka)*
|                    |           |                  | `arn:aws:controltower:ap-northeast-3::control/PKHZNEDFFYEC` |
|                    |           |                  | *Europe (Milan)*
|                    |           |                  | `arn:aws:controltower:eu-south-1::control/OOWBBBDUGZXPK` |
|                    |           |                  | *Africa (Cape Town)*
|                    |           |                  | `arn:aws:controltower:af-south-1::control/FAEROJSFBVFU` |
|                    |           |                  | *Middle East (Bahrain)*
|                    |           |                  | `arn:aws:controltower:me-central-1::control/CDQCCVMQOIDD` |
|                    |           |                  | *Israel (Tel Aviv)*
|                    |           |                  | `arn:aws:controltower:il-central-1::control/QCKSAFP00QKD` |
|                    |           |                  | *Europe (Zurich)*
|                    |           |                  | `arn:aws:controltower:eu-central-2::control/VULXBGHMFRCG` |
|                    |           |                  | *Europe (Spain)*
|                    |           |                  | `arn:aws:controltower:eu-south-2::control/LSFHGNPWFHOV` |
|                    |           |                  | *Asia Pacific (Hyderabad)*
|                    |           |                  | `arn:aws:controltower:ap-south-2::control/KJZFVGAUSQ` |
|                    |           |                  | *Middle East (UAE)*
<p>|                    |           |                  | <code>arn:aws:controltower:me-central-1::control/YFGVTIUSFTEA</code> |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.OPENSEARCH.PR.11 | • NIST 800-53 Rev 5 AC-4  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-23  
• NIST 800-53 Rev 5 SC-23(3)  
• NIST 800-53 Rev 5 SC-7(4)  
• NIST 800-53 Rev 5 SC-8  
• NIST 800-53 Rev 5 SC-8(1)  
• NIST 800-53 Rev 5 SC-8(2)  
• PCI DSS version 3.2.1 4.1 | • Encrypt data in transit | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/LSGSWJZPSQLN  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/XEOVAXSGANOW  
• US West (Oregon)  
arn:aws:aws:controltower:us-west-2::control/PRTVHUYCEBIA  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/JJJBKFBKOIFL  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/YBEBQCIZYYLYW  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/NRCZHBGZPNFI  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/RVUKNMPZHEOE  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/DDNSAGXFVZYE  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/RBAIGZPHECPA |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/AMIYCNJHXFZG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/BPRGFRWRLYQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/SHXZHXDYIIEM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/BBXSREYFRMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/IYYIWMMLYMGN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/CSBQQWECHTXL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ITXUJRWDZCIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/XIKCXGOTLKEJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/SVwDDTCGvTOR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/VZMEQZKMKZTD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/KZQCIJBAVBFU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4(26)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/JXKVVKUICIVI</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6(9)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/KXQXMXCMJACX</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td>• US West (Seattle) arn:aws:controltower:us-west-3::control/QYKSKQRQJKX</td>
</tr>
</tbody>
</table>

• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/LEHJKCAWOHPJ
• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/SKZXZATSLYXJ
• Europe (Zurich) arn:aws:controltower:eu-central-2::control/NZUIJJPXVQDJ
• Europe (Spain) arn:aws:controltower:eu-south-2::control/TZUJHBAATELA
• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/XOAOPZFFESJ
• Middle East (UAE) arn:aws:controltower:me-central-1::control/NXTCIIITQHWKW
• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/LUZFSXFQSKRQ
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST 800-53 Rev 5 AU-12</td>
<td>-</td>
<td>-</td>
<td>west-2::control/AOMKMLKJIIIH</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 AU-2</td>
<td>-</td>
<td>-</td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/GBORBUPXXVXWV</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 AU-3</td>
<td>-</td>
<td>-</td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/GIVZCHZYHVFF</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 AU-6(3)</td>
<td>-</td>
<td>-</td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ABSANFLGUHSE</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 AU-6(4)</td>
<td>-</td>
<td>-</td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/BPDFIRXUUC0O</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 CA-7</td>
<td>-</td>
<td>-</td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/YHCHFBWNQKEM</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7(9)</td>
<td>-</td>
<td>-</td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ABQEZJNCVZNTT</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SI-3(8)</td>
<td>-</td>
<td>-</td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/UIQWGTZTLAHXU</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SI-4(20)</td>
<td>-</td>
<td>-</td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/AADYUONVCYL</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SI-7(8)</td>
<td>-</td>
<td>-</td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/LXMWYMCWWQQDQ</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.1</td>
<td>-</td>
<td>-</td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/FLPLZPRKQSRW</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.2.1</td>
<td>-</td>
<td>-</td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/HIGDEERKHYKNS</td>
</tr>
</tbody>
</table>
| PCI DSS version 3.2.1 10.2.2 | - | - | South America (São Paulo) arn:aws:controltower:sao-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>east-1::control/VCPRGZXFHWH</td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/TJDUFYTASRZU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/MTPHRGMYMTNZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/GXDBIGPNJLKC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/PFESRXEPJLXF</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/YZFSNBZYNGBP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/SNIFQJTWZTDE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/JEYPGHUYMGGL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/FEJRRCAXKHDZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/QKHWLPCVKOEF</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/ITALJTQKWUNYO</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/GAEKXUENNOTD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.OPENSEARCH.PR.13 | • NIST 800-53 Rev 5 AC-2(4)  
• NIST 800-53 Rev 5 AC-4(26)  
• NIST 800-53 Rev 5 AC-6(9)  
• NIST 800-53 Rev 5 AU-10  
• NIST 800-53 Rev 5 AU-12  
• NIST 800-53 Rev 5 AU-2  
• NIST 800-53 Rev 5 AU-3  
• NIST 800-53 Rev 5 AU-6(3)  
• NIST 800-53 Rev 5 AU-6(4)  
• NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 SC-7(9)  
• NIST 800-53 Rev 5 SI-3(8)  
• NIST 800-53 Rev 5 SI-4(20)  
• NIST 800-53 Rev 5 SI-7(8)  
• PCI DSS version 3.2.1 10.1  
• PCI DSS version 3.2.1 10.2.1 | • Establish logging and monitoring | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
SFXALXZDGECS  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
USTIOXRNYFAV  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
XUNNABZCEQYZ  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
MHCOPBGLGBET  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
LIQKLRNJRVLN  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
MSQLW0ZPLNJR  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/  
ZSJSCMIFQYHQ  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/  
ZOCELNRKWSQC |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| • PCI DSS version 3.2.1 10.2.2 | | | • Europe (London)  
arn:aws:controltower:eu-west-2::control/EGIAPDZCHM0G |
| • PCI DSS version 3.2.1 10.2.3 | | | • Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/SLMGEZLLTGP0T |
| • PCI DSS version 3.2.1 10.2.4 | | | • Asia Pacific (Mumbai)  
arn:aws:controltower:ap-south-1::control/XRXYSFHHVRGW |
| • PCI DSS version 3.2.1 10.2.5 | | | • Asia Pacific (Seoul)  
arn:aws:controltower:ap-northeast-2::control/BOWBHCDYIKIV |
| • PCI DSS version 3.2.1 10.2.7 | | | • Asia Pacific (Tokyo)  
arn:aws:controltower:ap-northeast-1::control/IHQJBYGOYRNJ |
| • PCI DSS version 3.2.1 10.3.1 | | | • Europe (Paris)  
arn:aws:controltower:eu-west-3::control/WWWKEBANHSYO |
| • PCI DSS version 3.2.1 10.3.2 | | | • South America (São Paulo)  
arwn:aws:controltower:sa-east-1::control/CTHPVZJBDJ |
| • PCI DSS version 3.2.1 10.3.3 | | | • US West (N. California)  
arwn:aws:controltower:us-west-1::control/PMWAVYJPMSHES |
| • PCI DSS version 3.2.1 10.3.4 | | | • Asia Pacific (Hong Kong)  
arwn:aws:controltower:ap-east-1::control/DRJJPMHHMDLX |
| • PCI DSS version 3.2.1 10.3.5 | | | • Asia Pacific (Jakarta)  
arwn:aws:controltower:ap-southeast-3::control/CRIYVMVAJETX |
| • PCI DSS version 3.2.1 10.3.6 | | | • Asia Pacific (Osaka)  
arwn:aws:controltower:ap-northeast-3::control/OCIFQRYXYVJRZ |
| | | | • Europe (Milan)  
arwn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.OPENSEARCH.PR.14</td>
<td><strong>NIST 800-53 Rev 5 CP-10</strong>&lt;br&gt;<strong>NIST 800-53 Rev 5 CP-6(2)</strong></td>
<td>• Improve availability</td>
<td>• <strong>US East (N. Virginia)</strong>&lt;br&gt;arn:aws:controltower:us-east-1::control/GXAUYPCLBCBS&lt;br&gt;• <strong>US East (Ohio)</strong>&lt;br&gt;arn:aws:controltower:us-east-1::control/GXAUYPCLBCBS</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-36</td>
<td></td>
<td></td>
<td>east-2::control/JKMVKOYUMAPQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/CFWVMYGFBCAX</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/BHKRHJKOIAARA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/HYENHQKNDJCT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/XBNRHLHSYAQW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/SBAGREZDUTQH</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/OMMRADBYMNJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/BAKGLCSOQAJV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/POFYFDIMHQKH</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/AVEPZTWUQOHV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/00ELHVJWNSQU</td>
</tr>
</tbody>
</table>
| • NIST 800-53 Rev 5 SI-13(5) | | | Europe (Paris) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>west-3::control/GVQQKJDZIBUM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/FCOHMNGYYZGMQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ERMHCHCHXXQXM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/UBVVZQBVUCCN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/DSVSBRMWZIKQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/VXVTTKTVHATX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/HZEGOTZOHZQB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/COTPYWONNCWV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/ZDYNQADBTKIZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/GPETYRREPRRN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/DFGBJORYYMNY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/PWYCKZXADUQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/DYLEDFCIQYCR</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/JIEHZKZEXREPA</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-5</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/AXPLMDKRP5RX</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/GTQNOHLMVWMV</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/0RFJWTPQGSZ5</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eucentral-1::control/CYLADYYNHHZ5X</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/ SXSLJJAOTGMY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ SRFWZ3MTQRRZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ XMRBS3AKYPQW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ TJJENZZFEAXL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ CSJDXWLKOSQP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ NCZ0CAQXAZCZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ EFX0XQWWSZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ CFZPGDJXQJGW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ EBPSQUNFBDRL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ OHTDBHWVHJKO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ RUVAAYRTPBWD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Asia Pacific (Osaka) | arn:aws:controltower:ap-
## Control Identifiers, Frameworks, Control Objectives, and Control API Identifiers, by Region

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
<td>east-1::control/ULWLGPROCXUE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 IA-5(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-12(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-23(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US East (Ohio)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/WICEUAFLUNHP</td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/SAWLZWOPSRIB</td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/BYMWNKEJVBPA</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/SUIONSKYSCOB</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/RZWLHNJDKWJW</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:europe-central-1::control/RFZMHDDCATDV</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:europe-west-1::control/VTQFJKCNUPTA</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:europe-west-2::control/QMPILNQUJKOC</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:europe-north-1::control/YUIUSCIOMUDH</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/KZTACTIDLFA</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/QUFLTZGKIHYN</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo) | | | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeast-1::control/UYTWQU1JSXPK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/RHPSFIDONVGY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/OHTIGEPUEYQN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/AKOKGMZDNDBH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/HQXCYQHUHXDGA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/BCISWMGTUIEK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/PKHUQJEJCLW5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/JSJWBUBBPJFBG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/UYWVKVDKLLPS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/IWUPTYIRKSTI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/UFPFOLWQWQMA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/WTOJCTPLOFOI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## CT.OPENSEARCH.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.OPENSEARCH.PR.2 | • NIST 800-53 Rev 5 AC-21  
• NIST 800-53 Rev 5 AC-3  
• NIST 800-53 Rev 5 AC-3(7)  
• NIST 800-53 Rev 5 AC-4  
• NIST 800-53 Rev 5 AC-4(21)  
• NIST 800-53 Rev 5 AC-6  
• NIST 800-53 Rev 5 SC-7  
• NIST 800-53 Rev 5 SC-7(11)  
• NIST 800-53 Rev 5 SC-7(16)  
• NIST 800-53 Rev 5 SC-7(20)  
• NIST 800-53 Rev 5 SC-7(21)  
• NIST 800-53 Rev 5 SC-7(3) | • Limit network access | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/VTLIUXDAVRDR  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/XEVFCYEQTWZ  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/IHEMPFCKBBEK  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/XMRAHLUXZRT  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/GYGEIIYYZNL  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/NXUQGVFMOHAG |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/ACUNINHQNGLI</td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7(9)</td>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/LPUSJOMLH4GCV</td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.2.1</td>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/WFQXRFHOQCU</td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.3</td>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/QLSJKVLPLMMO</td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.3.1</td>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/TSAAUQ5ILYSH</td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.3.2</td>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/UCLVYXAVOAHM</td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.3.4</td>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/TJJGFRBZRZSV</td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.3.6</td>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/NEEJCATZEIMW</td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.3.9</td>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/BGWSRBPMEUSUR</td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.3.10</td>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/IFNDKBNUSCKD</td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.3.11</td>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/BXKDBCNEKUKQ</td>
<td></td>
</tr>
</tbody>
</table>
| PCI DSS version 3.2.1 1.3.12 | Asia Pacific (Jakarta) | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>southeast-3::control/YCCNRZZHQRUAU</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/CQLGRRFXIEID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/NPLWOOSVWQN5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/WVG1WATQXNV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/ODWPJGSCURQW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/TSMMXHLKSOGU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/QWMIYWQWWHV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/CERUJQSWOLEG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/VOYUOLERDHMA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/OZVNVIANAQY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/NLYEGMAOFTAD</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.3</td>
<td>• NIST 800-53 Rev 5 AC-4&lt;br&gt;• NIST 800-53 Rev 5 SC-13&lt;br&gt;• NIST 800-53 Rev 5 SC-23&lt;br&gt;• NIST 800-53 Rev 5 SC-23(3)&lt;br&gt;• NIST 800-53 Rev 5 SC-7(4)&lt;br&gt;• NIST 800-53 Rev 5 SC-8&lt;br&gt;• NIST 800-53 Rev 5 SC-8(1)&lt;br&gt;• NIST 800-53 Rev 5 SC-8(2)&lt;br&gt;• PCI DSS version 3.2.1 4.1</td>
<td>• Encrypt data in transit</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/HLPMHXZUFA</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/UGGNPNQICGZQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/WZFMVXTTLCMFR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/MTARCONVTLOX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/JFSVGFJCHFQZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/PUPHEVPARPK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/GZFIOCNJXQLT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/QLYZGPDANYDA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/RAGNOCSYSLJR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/HQFAQRQRUCY5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/VHMQVLLVZKWN0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/PTVICLGFUOIP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## CT.OPENSEARCH.PR.4

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4(26)</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ZJUCJJOESN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6(9)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/AKGAYZTWFZI</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/DOPPXRVAYVN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Europe (Zurich)** arn:aws:controltower:eu-central-2::control/HOWAJKATTUKZ
- **Europe (Spain)** arn:aws:controltower:eu-south-2::control/AFJCHTBCRUSW
- **Asia Pacific (Hyderabad)** arn:aws:controltower:ap-south-2::control/HFBWABBDQUGE
- **Middle East (UAE)** arn:aws:controltower:me-central-1::control/FPCUMXNOKTVA
- **Asia Pacific (Melbourne)** arn:aws:controltower:ap-southeast-4::control/GRSDIFUYIMFN
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
<td>southeast-2::control/UVAXZWGNLSOM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/NOEXBYDHYFLN</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/QWERPPLZPCHT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/VPWNXJISTFOH</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/YZZBKRFXBVMZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/KLCYSVFSKGB0</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/EJLRJLUVBNHN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ZYBIFAAQQZB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/NEMEBLSVQDQW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/UHBZCFPBQDRL</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/AJUPFJHAONI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.7</td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/EWSFLJTIRUHU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>[arn:aws:controltower:ap-east-1::control/] DJVTCGPZYSLR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>[arn:aws:controltower:ap-southeast-3::control/] TOLOKFEEGTJH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>[arn:aws:controltower:ap-northeast-3::control/] ZVKTNWLSWKUC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>[arn:aws:controltower:eu-south-1::control/] BQZZSSJVRNHD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>[arn:aws:controltower:af-south-1::control/] JCZCTQBRLIFS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>[arn:aws:controltower:me-south-1::control/] WZCAZXFJWWFV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>[arn:aws:controltower:il-central-1::control/] GFKZAMSBYHGQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>[arn:aws:controltower:eu-central-2::control/] LZHEPFFMOGFV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>[arn:aws:controltower:eu-south-2::control/] ZKDDJNJNGEET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>[arn:aws:controltower:ap-south-2::control/] RHZECQRAAAZW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>[arn:aws:controltower:me-central-1::control/] PGSJHVI0VEJV</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Asia Pacific (Melbourne) | [arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.OPENSEARCH.PR.5 | • NIST 800-53 Rev 5 AC-2(4)  
• NIST 800-53 Rev 5 AC-4(26)  
• NIST 800-53 Rev 5 AC-6(9)  
• NIST 800-53 Rev 5 AU-10  
• NIST 800-53 Rev 5 AU-12  
• NIST 800-53 Rev 5 AU-2  
• NIST 800-53 Rev 5 AU-3  
• NIST 800-53 Rev 5 AU-6(3)  
• NIST 800-53 Rev 5 AU-6(4)  
• NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 SC-7(9)  
• NIST 800-53 Rev 5 SI-3(8)  
• NIST 800-53 Rev 5 SI-4(20)  
• NIST 800-53 Rev 5 SI-7(8)  
• PCI DSS version 3.2.1 10.1  
• PCI DSS version 3.2.1 10.2.1  
• PCI DSS version 3.2.1 10.2.2  
• PCI DSS version 3.2.1 10.2.3  
• PCI DSS version 3.2.1 10.2.4 | • Establish logging and monitoring | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/KYAFRIXOLJCM  
• US East (Ohio) arn:aws:controltower:us-east-2::control/LVNKFQFKMTWR  
• US West (Oregon) arn:aws:controltower:us-west-2::control/CNSIZGNVLCLN  
• Canada (Central) arn:aws:controltower:ca-central-1::control/WZANTGTEGLMA  
• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ORIKZBBSZRB  
• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/SVOHVTULZGK  
• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/TWOQLVIYAVG  
• Europe (Ireland) arn:aws:controltower:eu-west-1::control/QEEZPBMZYQFN  
• Europe (London) arn:aws:controltower:eu-west-2::control/YCOUGYAZOQNA  
• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/GLKQJFW5KVCH |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| • PCI DSS version 3.2.1 10.2.5<br>• PCI DSS version 3.2.1 10.2.7<br>• PCI DSS version 3.2.1 10.3.1<br>• PCI DSS version 3.2.1 10.3.2<br>• PCI DSS version 3.2.1 10.3.3<br>• PCI DSS version 3.2.1 10.3.4<br>• PCI DSS version 3.2.1 10.3.5<br>• PCI DSS version 3.2.1 10.3.6 | Asia Pacific (Mumbai)<br>Asia Pacific (Seoul)<br>Asia Pacific (Tokyo)<br>Europe (Paris)<br>South America (Rio)<br>US West (N. California)<br>Asia Pacific (Hong Kong)<br>Asia Pacific (Jakarta)<br>Asia Pacific (Osaka)<br>Europe (Milan)<br>Africa (Cape Town)<br>Middle East (Bahrain) | Arn:aws:controltower:ap-south-1::control/FRHDOEXXBOWQ<br>Arn:aws:controltower:ap-northeast-2::control/JCCUGYQNCITY<br>Arn:aws:controltower:ap-northeast-1::control/ZBUOY1WGUJZP<br>Arn:aws:controltower:eu-west-3::control/USWETIYXXFTH<br>Arn:aws:controltower:sa-east-1::control/ZJMOCDQOAMSQ<br>Arn:aws:controltower:us-west-1::control/EIHWFSQEDMQS<br>Arn:aws:controltower:ap-east-1::control/DNGAAKXQFDQV<br>Arn:aws:controltower:ap-southeast-3::control/PWUNDFXSUYAY<br>Arn:aws:controltower:ap-northeast-3::control/VXAOMOWSCIBP<br>Arn:aws:controltower:eu-south-1::control/JVKVHVFYVEXB<br>Arn:aws:controltower:af-south-1::control/IBPRSOCDFQRG<br>Arn:aws:controltower:me-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.OPENSEARCH.PR.6 | • NIST 800-53 Rev 5 CP-10  
• NIST 800-53 Rev 5 CP-6(2)  
• NIST 800-53 Rev 5 SC-36  
• NIST 800-53 Rev 5 SC-5(2)  
• NIST 800-53 Rev 5 SI-13(5) | • Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/ AIBFRQCHGJ00  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/ LXDACFXTM10S  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/ ENATZEFXPYYM  
• Canada (Central)  
arn:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>central-1::control/ATEKPCKHHHYQ</td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/UTDGLWWSQDEU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/VJKQXGOJJPYX</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eucentral-1::control/EMJYVYDQOQRX</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/QMRIWQGLWTKC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/EREYWUMMXCYD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/WSBITCGHGEDK</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/HKMNHQCBPFBII</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/BQNHZOCGNLIA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ULUJPIODNXMQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/HRQELGWRN0FE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:saeast-1::control/LIRWXACRJOHY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• US West (N. California)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/DNMZNOAPKWK0</td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/NHTZGOYISWAH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/HIZSJVEOIWXP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/WFTCZPQQJGQN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/PYSBFVWYCMUAU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/ARTUADLEDUAQ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/JVWSMFEROGHX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/XKCISKPFWCSK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/VPIIPJ0BXEET</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/FECUISCKYVVV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/OCLOCVAYTOOI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/GRIXGVMZZGTT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------</td>
<td>--------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.7</td>
<td>NIST 800-53 Rev 5 CP-10</td>
<td>Improve availability</td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/YZEZWADHNHLT</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-36</td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/AIO0YWFDGTZ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/WKDTZIOQBFDF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/MDJZFKJKKFMG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/BEKZJRWESGED</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/YXTTBALIVPSX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/INNDAQGPPHVVC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/UHQAJJBDQDDJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/CGY0VDBJAZLY</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/GFZRKVAQNTNM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/SJEVTTRVISWQO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/LDJIZXCUADJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/GKHBCXGNPZMK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/THXOAXRJOLPA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/ITGIWIHHKFYG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/CACOFIMRPBKI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/O3FEGCMAQKJ0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/BQWPTLYFLVJTT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/BAKHBLHLOYKAA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/ASN5DRVZSYXT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.OPENSEARCH.PR.8</td>
<td>NIST 800-53 Rev 5 AC-17(2)</td>
<td>Encrypt data in transit</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/OHXMUXUWTJTT</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/WJYDNSSTJUZJ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 IA-5(1)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/HQCMQZJWZHH</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-12(3)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/HQCMQZJWZHH</td>
</tr>
</tbody>
</table>

- Middle East (Bahrain) arn:aws:controltower:me-south-1::control/HILHCVHZRUJ
- Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/HOJNWTFCAPQJ
- Europe (Zurich) arn:aws:controltower:eu-central-2::control/BQNTMFNFOIRL
- Europe (Spain) arn:aws:controltower:eu-south-2::control/GUCQQGCDGDNS
- Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/TTPHKLHOVRAJ
- Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/YCWSPDDLEEDG

2004
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td></td>
<td>west-2::control/BAGLVGGLRFN</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-23</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/PKOEZEOEQMU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-23(3)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/RUFUVUMOQXR</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/CLXJODNHERHU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ZZTFUHNJWNBG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(1)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/YKVRVXHEMUIB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/NOACZITNOFIL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/XUOYURYPSGTI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/EJEEGAXFUIUQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ALICRWQMIXIKK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/AJMLARVDOFYW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/RTMLTTPISDQU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>SKGWRAFAEEAUZ</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td>- US West (N. California)</td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td></td>
<td>- US West (N. California)</td>
</tr>
<tr>
<td>urn:aws:controltower:us-west-1::control/</td>
<td></td>
<td></td>
<td>UMATVTJIVRDFT</td>
</tr>
<tr>
<td>• Asia Pacific (</td>
<td></td>
<td></td>
<td>- Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td>Hong Kong)</td>
<td></td>
<td></td>
<td>- Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td>urn:aws:controltower:ap-east-1::control/</td>
<td></td>
<td></td>
<td>AJFWGMGACTIU</td>
</tr>
<tr>
<td>• Asia Pacific (</td>
<td></td>
<td></td>
<td>- Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td>Jakarta)</td>
<td></td>
<td></td>
<td>- Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td>urn:aws:controltower:ap-southeast-3::control/</td>
<td></td>
<td></td>
<td>WHKQJXKELYJR</td>
</tr>
<tr>
<td>• Asia Pacific (</td>
<td></td>
<td></td>
<td>- Asia Pacific (Osaka)</td>
</tr>
<tr>
<td>Osaka)</td>
<td></td>
<td></td>
<td>- Asia Pacific (Osaka)</td>
</tr>
<tr>
<td>urn:aws:controltower:ap-northeast-3::control/</td>
<td></td>
<td></td>
<td>MDWFN2OJOJXSM</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>- Europe (Milan)</td>
</tr>
<tr>
<td>urn:aws:controltower:eu-south-1::control/</td>
<td></td>
<td></td>
<td>NSOLJEXJZYNL</td>
</tr>
<tr>
<td>• Africa (Cape</td>
<td></td>
<td></td>
<td>- Africa (Cape Town)</td>
</tr>
<tr>
<td>Town)</td>
<td></td>
<td></td>
<td>- Africa (Cape Town)</td>
</tr>
<tr>
<td>urn:aws:controltower:af-south-1::control/</td>
<td></td>
<td></td>
<td>VVGQWQKJFVYY</td>
</tr>
<tr>
<td>• Middle East (</td>
<td></td>
<td></td>
<td>- Middle East (Bahrain)</td>
</tr>
<tr>
<td>Bahrain)</td>
<td></td>
<td></td>
<td>- Middle East (Bahrain)</td>
</tr>
<tr>
<td>urn:aws:controltower:me-south-1::control/</td>
<td></td>
<td></td>
<td>YVHFJLXVZJFJ</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>- Israel (Tel Aviv)</td>
</tr>
<tr>
<td>urn:aws:controltower:il-central-1::control/</td>
<td></td>
<td></td>
<td>XUZUWYQWQFJFI</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>- Europe (Zurich)</td>
</tr>
<tr>
<td>urn:aws:controltower:eu-central-2::control/</td>
<td></td>
<td></td>
<td>ZJNNFNLFXDSTY</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>- Europe (Spain)</td>
</tr>
<tr>
<td>urn:aws:controltower:eu-south-2::control/</td>
<td></td>
<td></td>
<td>GTNCQJERTNUR</td>
</tr>
<tr>
<td>• Asia Pacific (</td>
<td></td>
<td></td>
<td>- Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td>Hyderabad)</td>
<td></td>
<td></td>
<td>- Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td>urn:aws:controltower:ap-south-2::control/</td>
<td></td>
<td></td>
<td>AVGRWYMXRRX</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.OPENSEARCH.PR.9 |           | Encrypt data at rest | US East (N. Virginia) arn:aws:controltower:useast-1::control/KHZCJMYXGFVCV  
|                    |           |                  | US East (Ohio) arn:aws:controltower:useast-2::control/CZMXNAENBDTR  
|                    |           |                  | US West (Oregon) arn:aws:controltower:uswest-2::control/ZIBAOFHKZHQY  
|                    |           |                  | Canada (Central) arn:aws:controltower:ccentral-1::control/IXYPCNJPAPVYP  
|                    |           |                  | Asia Pacific (Sydney) arn:aws:controltower:apsoutheast-2::control/GLPWOLQVEQJQ  
|                    |           |                  | Asia Pacific (Singapore) arn:aws:controltower:apsoutheast-1::control/UCOLFZDGG3JQQ  
|                    |           |                  | Europe (Frankfurt) arn:aws:controltower:ecentral-1::control/MGUGUPVHUC  
|                    |           |                  | Europe (Ireland) arn:aws:controltower:euwest-1::control/ICDKMTFLINVG  
|                    | NIST 800-53 Rev 5 CA-9(1) |  
|                    | NIST 800-53 Rev 5 CM-3(6) |  
|                    | NIST 800-53 Rev 5 SC-13 |  
|                    | NIST 800-53 Rev 5 SC-28 |  
|                    | NIST 800-53 Rev 5 SC-28(1) |  
|                    | NIST 800-53 Rev 5 SI-7(6) |  
|                    | PCI DSS version 3.2.1 3.4 |  
|                    | PCI DSS version 3.2.1 8.2.1 |  
|                    | Middle East (UAE) arn:aws:controltower:me-central-1::control/EDRAKQMAORKE |  
|                    | Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/CJXLXUE3MVCD |  

2007
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/NUKBIVYHBXOKQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/GKDVQZT0HFQG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td>(Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/GWVTUMUOXNCX</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td>(Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/LITFPWDWFWM</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td>(Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/QXOGTDBDLUJI</td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/CUUDCPZYQEDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America</td>
<td>(São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/DBOUBKGZADQE</td>
<td></td>
</tr>
<tr>
<td>• US West (N.</td>
<td>California)</td>
<td>arn:aws:controltower:us-west-1::control/PHDGPGPDNLRZ</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td>(Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/AASYBLNJTFFY</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td>(Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/LYXBCHEEZPVN</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td>(Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ZQAMBSNTXDIJ</td>
<td></td>
</tr>
</tbody>
</table>
| • Europe (Milan)   | arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.RDS.PR.1         | • CIS AWS Benchmark 1.4 2.3.1  
                     • NIST 800-53 Rev 5 AC-2(4) | • Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
CIPCPCUPIFSX  
• US East (Ohio)  
ar:aws:controltower:us-east-2::control/  
NPIGGPFBTJ |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td>east-2::control/ CPBJPVIEHMMU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ BXNPKKZKAPUL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(26)</td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ QFAFUIKDIHXA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(9)</td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ XFYDDQMPBSV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-10</td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ PBWERHRRPVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-12</td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ NGZCEDBORKNM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-2</td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ ISZORRYWJRHN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-3</td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ DKTUUXJADDDB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ DQYDALNZIZZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ HZBEHSXYTYJP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ IQZLHKCZPRDQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ GYEHZ0AWYUUR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
<td>2010</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/RAGAKMUCIMRJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/TNHSQNYQQZFT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/RAXPKGFBBHIP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/PHKJMGXZWMWH</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/HUPJOPPYSUGA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/ACFMYSTVSRBK</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/GICLLRYRAVTC</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(5)</td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/YDIFCPCASQLB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/UUBOTNXZHMGL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/YLJBMFVYQGIP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/ZRDOEHVIOCLY</td>
</tr>
</tbody>
</table>
| • NIST 800-53 Rev 5 SI-3(8) | | | • Europe (Spain) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.RDS.PR.10</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Protect configurations</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/TDLAWEVSVJFW</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>south-2::control/UMPYPULZBKMW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/JEHZ0TCTRSDL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE) arn:aws:controltower:me-central-1::control/YGNVONVKFHAAS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/BPVKEWVTUWLR</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CT.RDS.PR.10**

- PCI DSS version 3.2.1 10.2.3
- PCI DSS version 3.2.1 10.2.4
- PCI DSS version 3.2.1 10.2.5
- PCI DSS version 3.2.1 10.2.6
- PCI DSS version 3.2.1 10.2.7
- PCI DSS version 3.2.1 10.3.1
- PCI DSS version 3.2.1 10.3.2
- PCI DSS version 3.2.1 10.3.3
- PCI DSS version 3.2.1 10.3.4
- PCI DSS version 3.2.1 10.3.5
- PCI DSS version 3.2.1 10.3.6
- PCI DSS version 3.2.1 11.5
- PCI DSS version 3.2.1 2.1
- PCI DSS version 3.2.1 2.2
- PCI DSS version 3.2.1 2.2.2
- PCI DSS version 3.2.1 2.4
- PCI DSS version 3.2.1 3.4
- PCI DSS version 3.2.1 8.2.1

- NIST 800-53 Rev 5 CA-9(1)
- NIST 800-53 Rev 5 CM-2

- Protect configurations
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/MIIGPMTVJKED</td>
</tr>
<tr>
<td>CM-2(2)</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/YISDVYQBVSTR</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/DVXCTDPQANLR</td>
</tr>
<tr>
<td>2.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ OVTFQMGWPNHM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ QESQSXDHEYUX</td>
</tr>
<tr>
<td>2.4</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ONNOHNAHLBAH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/LYZSDLIUYNXH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/EAHGNZJQRUJY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ OLNHXEKJZGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/CGCCLNZVZFEY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/SECLZXVIEZ0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/LXVUL0YZQA0H</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/IXRCLTJDINER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/NNOKBENVMLPI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/NZKBJFTZDXTE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/ZNKDVODBDAVE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/UZRSPFWZQTYU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/UXAVTPDBVFPF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/CKJESAIGJNTU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/LBQMOPFWQPXZ5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/QFKZNHJKTIYZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/LXDGZHIMLLLKF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/TVWQAXDABKP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Spain)</td>
</tr>
</tbody>
</table>
|                    |           |                  | arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.RDS.PR.11        |           | Limit network access | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/SBVAEBQZFXKJ  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/QPGIONOVKEPL  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/SAVOXDMXWNV  
• Canada (Central)  
ar:n:aws:controltower:ca-central-1::control/HTVHIQWPULEZ  
• Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/OXXJQNOBPMWSI  
• Asia Pacific (Singapore)  
ar:n:aws:controltower:ap-southeast-1::control/RLRMHNBZWMZ  
• Europe (Frankfurt)  
ar:n:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>central-1::control/QGPVEJMHJHPN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/QLIQBXFPMWQG</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/QPIWXZNGMLE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/EOHTLNPNLNAHI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ZZRRQNGFFMLK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/IFJMWZQBBHUIV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/BLRCFNBJELWJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/OCJRITLFUKTI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/FLMYCPVPBXRI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/NAWSDEKFNZX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/FXBSIFWKQXOD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/NHKXBYQLKDDS</td>
</tr>
</tbody>
</table>

2016
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/HRHVBEGGVKUN</td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/NBWURONLVXRT</td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/DIVKMUVKJCWC</td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/NWDRVUJ1WSMZ</td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/KZYGQTBLAIMC</td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/XBPNREGMWEXZ</td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/AHXQJSVNVUIS</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/JOXYQPQMCAXO</td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/JNEGMI0FANOA</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Melbourne)</td>
<td>arn:aws:controltower:ap-southeast-4::control/ZFNHHDPYTJBL</td>
<td></td>
</tr>
</tbody>
</table>
## Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
CT.RDS.PR.12 | • NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 SI-2  
• PCI DSS version 3.2.1 11.5 | • Prepare for incident response | • **US East (N. Virginia)** arn:aws:controltower:us-east-1::control/UDBQCAZJXEAD  
• **US East (Ohio)** arn:aws:controltower:us-east-2::control/WGTGNSNVQASQ  
• **US West (Oregon)** arn:aws:controltower:us-west-2::control/SHDPEBVLYNBQ  
• **Canada (Central)** arn:aws:controltower:ca-central-1::control/RVAPYALZJHJLY  
• **Asia Pacific (Sydney)** arn:aws:controltower:ap-southeast-2::control/VQTXJCSDCVNL  
• **Asia Pacific (Singapore)** arn:aws:controltower:ap-southeast-1::control/ILUJFEPHTBGC  
• **Europe (Frankfurt)** arn:aws:controltower:eu-central-1::control/ORSIZYRJNQJK  
• **Europe (Ireland)** arn:aws:controltower:eu-west-1::control/XSWQZJIDXEXX  
• **Europe (London)** arn:aws:controltower:eu-west-2::control/WVSJOPRGOHET  
• **Europe (Stockholm)** arn:aws:controltower:eu-north-1::control/XLUMDEETUGPN  
• **Asia Pacific (Mumbai)** arn:aws:controltower:ap-south-1::control/SKNYYPGEGZZE
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/LXSLBPQPTPF</td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/LXSLBPQPTPF</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/WIIXHZHCJRQ</td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/WIIXHZHCJRQ</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/BEHCRCSQHEY</td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/BEHCRCSQHEY</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ZHQOPQLAGHIW</td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/ZHQOPQLAGHIW</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ECVBIUCVKDHD</td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/ECVBIUCVKDHD</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/GBNYUHASXVMY</td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/GBNYUHASXVMY</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/FILLVTGDVMRA</td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/FILLVTGDVMRA</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/GVLKLASFFMRA</td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/GVLKLASFFMRA</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/NMXEXSCYFMMR</td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/NMXEXSCYFMMR</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/CNDUIYYWVZVY</td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/CNDUIYYWVZVY</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-</td>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.RDS.PR.13        |           | Improve availability | US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/YPJCCMKZJPTP  
US East (Ohio) 
arnt:aws:controltower:us-east-2::control/LMAWCCZTMZMW  
US West (Oregon) 
arnt:aws:controltower:us-west-2::control/MZGZCGQQSWRE  
Canada (Central) 
arnt:aws:controltower:ca-central-1::control/WAJLGTNDLJS  
Asia Pacific (Sydney) 
arnt:aws:controltower:ap-southeast-4::control/SZBDICZIVCP5  
Europe (Zurich) 
arnt:aws:controltower:eu-central-2::control/SIIEASJGDLTH  
Europe (Spain) 
arnt:aws:controltower:eu-south-2::control/SSMTAR00YQXY  
Asia Pacific (Hyderabad) 
arnt:aws:controltower:ap-south-2::control/SKKORLKAZAOP  
Middle East (UAE) 
arnt:aws:controltower:me-central-1::control/GYCLJLXZUCKP  
Asia Pacific (Melbourne) 
arnt:aws:controltower:ap-southeast-4::control/SZBDICZIVCP5  
NIST 800-53 Rev 5 CA-9(1)  
NIST 800-53 Rev 5 CM-2  
NIST 800-53 Rev 5 CM-2(2)  
NIST 800-53 Rev 5 CM-3  
NIST 800-53 Rev 5 SC-5(2)  
PCI DSS version 3.2.1 2.2  

2020
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>southeast-2::control/CWTDADVXUVNS</td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ZQVRGYMNSXHA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/XVPZAGNVDDKT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ZTKNQGKSPGTR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/VSSGLB0ZRHRD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/IIMGJHZTBAFN</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/PKVINHPDQTMI</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/BFSGZKNINMRH</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ANNPBISWZVNU</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/UFCJQENIQLML</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/XVXISEOGBKUFM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/DOGCTOSZOKKF</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/OPTPMAIZQHQV</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/TCPYXNOZETYP</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/CEVYQIDVTPJR</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/UPMYISEYCVNAV</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/WQOZJPDGWHXW</td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/RWLYEFWQHCJ</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/WWCJGJPMADDZ</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/OXHET0CAIGYR</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/PXZORCGCQERT</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Hyderabad)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/BIZBTULXELSX</td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(UAE)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/ZHHLWPFNMOOS</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| (Melbourne)        |           |                  | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.RDS.PR.14        |           | • Establish logging and monitoring | • US East (N. Virginia) **arn:**aws:controltower:us-east-1::control/WWWIIFC0KJQR
|                    |           |                   | • US East (Ohio) **arn:**aws:controltower:us-east-2::control/LMBFAGECJTQZ
|                    |           |                   | • US West (Oregon) **arn:**aws:controltower:us-west-2::control/FGOSLUOXFYBB
|                    |           |                   | • Canada (Central) **arn:**aws:controltower:ca-central-1::control/AUSMNRMJUJF
|                    |           |                   | • Asia Pacific (Sydney) **arn:**aws:controltower:ap-southeast-2::control/HSIZECNPMUMR
|                    |           |                   | • Asia Pacific (Singapore) **arn:**aws:controltower:ap-southeast-1::control/DLUNVYXZCXARR
|                    |           |                   | • Europe (Frankfurt) **arn:**aws:controltower:eu-central-1::control/FWIYAMLIPWZL
|                    |           |                   | • Europe (Ireland) **arn:**aws:controltower:eu-west-1::control/UHEVILPYODIR
|                    |           |                   | • Europe (London) **arn:**aws:controltower:eu-west-2::control/DMXHIYJDVWWS
|                    |           |                   | • Europe (Stockholm) **arn:**aws:controltower:eu-north-1::control/RMQBHTEVKHEX

*Control identifier: CT.RDS.PR.14
*Framework: NIST 800-53 Rev 5
*NIST 800-53 Rev 5 AC-2(4)
*NIST 800-53 Rev 5 AC-4(26)
*NIST 800-53 Rev 5 AC-6(9)
*NIST 800-53 Rev 5 AU-10
*NIST 800-53 Rev 5 AU-12
*NIST 800-53 Rev 5 AU-2
*NIST 800-53 Rev 5 AU-3
*NIST 800-53 Rev 5 AU-6(3)
*NIST 800-53 Rev 5 AU-6(4)
*NIST 800-53 Rev 5 CA-7
*NIST 800-53 Rev 5 SC-7(10)
*NIST 800-53 Rev 5 SC-7(9)
*NIST 800-53 Rev 5 SI-3(8)
*NIST 800-53 Rev 5 SI-4(20)
*NIST 800-53 Rev 5 SI-7(8)
*PCI DSS version 3.2.1 10.1
*PCI DSS version 3.2.1 10.2.1
*PCI DSS version 3.2.1 10.2.2
*PCI DSS version 3.2.1 10.2.3
*NIST 800-53 Rev 5 AU-10
*NIST 800-53 Rev 5 AU-12
*NIST 800-53 Rev 5 AU-2
*NIST 800-53 Rev 5 AU-3
*NIST 800-53 Rev 5 AU-6(3)
*NIST 800-53 Rev 5 AU-6(4)
*NIST 800-53 Rev 5 CA-7
*NIST 800-53 Rev 5 SC-7(10)
*NIST 800-53 Rev 5 SC-7(9)
*NIST 800-53 Rev 5 SI-3(8)
*NIST 800-53 Rev 5 SI-4(20)
*NIST 800-53 Rev 5 SI-7(8)
*PCI DSS version 3.2.1 10.1
*PCI DSS version 3.2.1 10.2.1
*PCI DSS version 3.2.1 10.2.2
*PCI DSS version 3.2.1 10.2.3

2023
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td>• PCI DSS version 3.2.1 10.2.6</td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/NVAQMMWPIIPQ</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.2.7</td>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/XEFJXADQGRJR</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/BMFDQZNVAVYO</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/ADDPNHNPIOYV</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/RBLXDVKFFTW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/PIPVMVSYY0SQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/SAQZBRAUEBWQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/IGMNPKJRFZMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/DBPZUFKFKMP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/KCPAXGBPPFCI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:ap-south-1::control/GHZGSKQFCISF</td>
</tr>
</tbody>
</table>
| | | | • Middle East (Bahrain) arn:aws:controltower:me-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.RDS.PR.15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/BIMBKOKMEZXP</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/BRRGKLOMCWFQ</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 CM-8(1)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/NUDJWFBHWYF</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>central-1::control/KYNERGDEMOGQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/RZHJAVYJQGUI</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/IYNQLANVHZKU</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/NNIILIDPTJRP</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/HJRGQTAYQARY</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/OCSRNEWRPYWN</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/KTCCRRGSRUWGR</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/QJ00DGPKZYCB</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/YLCLBFJ3GFIZD</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/NJLPGDHZZWC0</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/BZCJR60KMRDB</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/MOKLHMPEKSI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/ZATLSZLRTTSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/DAKSTNVIEZPX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/YZEEBZWNKYW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/EXOHLMGVBSIW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/TCJCRQSDCNJK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/BJSYMTKHMSPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/MSITXUPZCMQA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/DUOALUMXEYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/IZYWDWEAIVN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/FTFOQPGTEYV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/QGBYVDCJSOMV</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------</td>
<td>-------------------------------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>CT.RDS.PR.16</td>
<td>• CIS AWS Benchmark 1.4 2.3.1</td>
<td>• Encrypt data at rest</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/FYVNKTVAUTPK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/DBQOQVLCRDJL</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/XRFPHUIOBFD</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ANFLBGZRYMCQ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/JCCMUPZAXMMW</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/CFYQQQEJAFFV</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(10)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/NEUJCNUIDIOU</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/WLPQBCCAWEG</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/WUYFXYDNPAJI</td>
</tr>
<tr>
<td></td>
<td>3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ MALKWFCLFDSJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ FWAZWNKRMETW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ JPOWMFDTTQPT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ ZNEMLOPSODLT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ ODZOMBQWAUBS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ TADJJIKRHQAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ MZZNTZGDUSZJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ NPBHBWQUQZGY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ VKMGPRLMNCBY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ OVSTMQMNPMI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/ JCRHCYEBEBBF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.RDS.PR.17       |           | • Prepare for incident response | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/SHNJOPOE2HZOMH  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/XKWPQGYXAMK  
• US West (Oregon)  
ar:n:aws:controltower:us-

- **Control identifier** | **Framework** | **Control objective** | **Control API identifiers, by Region** |
- CT.RDS.PR.17 | | • Prepare for incident response | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/SHNJOPOE2HZOMH  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/XKWPQGYXAMK  
• US West (Oregon)  
ar:n:aws:controltower:us-

- **Control identifier** | **Framework** | **Control objective** | **Control API identifiers, by Region** |
- CT.RDS.PR.17 | | • Prepare for incident response | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/SHNJOPOE2HZOMH  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/XKWPQGYXAMK  
• US West (Oregon)  
ar:n:aws:controltower:us-

- **Control identifier** | **Framework** | **Control objective** | **Control API identifiers, by Region** |
- CT.RDS.PR.17 | | • Prepare for incident response | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/SHNJOPOE2HZOMH  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/XKWPQGYXAMK  
• US West (Oregon)  
ar:n:aws:controltower:us-

- **Control identifier** | **Framework** | **Control objective** | **Control API identifiers, by Region** |
- CT.RDS.PR.17 | | • Prepare for incident response | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/SHNJOPOE2HZOMH  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/XKWPQGYXAMK  
• US West (Oregon)  
ar:n:aws:controltower:us-

- **Control identifier** | **Framework** | **Control objective** | **Control API identifiers, by Region** |
- CT.RDS.PR.17 | | • Prepare for incident response | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/SHNJOPOE2HZOMH  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/XKWPQGYXAMK  
• US West (Oregon)  
ar:n:aws:controltower:us-

- **Control identifier** | **Framework** | **Control objective** | **Control API identifiers, by Region** |
- CT.RDS.PR.17 | | • Prepare for incident response | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/SHNJOPOE2HZOMH  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/XKWPQGYXAMK  
• US West (Oregon)  
ar:n:aws:controltower:us-

- **Control identifier** | **Framework** | **Control objective** | **Control API identifiers, by Region** |
- CT.RDS.PR.17 | | • Prepare for incident response | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/SHNJOPOE2HZOMH  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/XKWPQGYXAMK  
• US West (Oregon)  
ar:n:aws:controltower:us-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>west-2::control/</td>
<td></td>
<td></td>
<td>YHBMVYOGUEDD</td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/ CBUKUNUOEILM</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/ TKAIVNUOZE0EP</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/ ODRTWYAHZKPB</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/ ASUNKGVFVPXY</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/ IRQZDEJQMXIL</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/ QAJDWDRVODFWX</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/ PBDUVQCOOKDV</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/ UBRPEHE0KHPO</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/ OUWNJTTOMXIL</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/ QOUOAILPPGLH</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/ WECTXDDEEGGMV</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>east-1:control/VIYXJSVPVORD</td>
<td>• US West (N. California) arn:aws:controltower:us-west-1:control/IZDMIZEKSPDS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1:control/EQZLUZVLFBMG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3:control/WUFEZ0IZMIXBA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3:control/MPZBVXOJLRKF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1:control/EQURNFTUNIJC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1:control/GZZZWRTHEHCNU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1:control/BUPIB3RNVOLH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1:control/WFCMRGL1GBCW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2:control/BE0E0CVQSEY0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2:control/UKQUAWKOJNPA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2:control/USENKGRNCKYN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.RDS.PR.18        | • NIST 800-53 Rev 5 CA-7  
                     • NIST 800-53 Rev 5 SI-2  
                     • PCI DSS version 3.2.1 11.5 | • Prepare for incident response | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
HMGIGLGDRLMZO |
|                     |           |                  | • US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
KRKPUXQSNVYD |
|                     |           |                  | • US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
GGCJMMNPCKNQG |
|                     |           |                  | • Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
MKVLPSPIEPIO |
|                     |           |                  | • Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
TXDOP00EOF0GJ |
|                     |           |                  | • Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
KOPCQKNKSYEUZ |
|                     |           |                  | • Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/  
HTOVAPFGMPCJ |
|                     |           |                  | • Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/  
RQCDURNMVVT5 |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/KMKTHWDNXIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/QZFGDRVZILTN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/YKUELCLADAOM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/YHSXDDXXWITQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/PFPVEOVUMUUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/GFMLOENIRNTC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/GLYKLGYIXELE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/SAJRANHECEJB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/THYPQJUMLHKT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/QNTSSYLVYSBR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/GXNWDDJVUIVA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Europe (Milan)    | arn:aws:controltower:eu-west-4::control/GI2X49T9Z5W }
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 11.5</td>
<td></td>
<td>east-2::control/QWXNKLRWSHXRQ</td>
<td></td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/XUPYDDLWHDCG</td>
<td></td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/PJX00IPSMBNA</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td>arn:aws:aws:controltower:ap-southeast-2::control/AKURSKMIRNMZ</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td>arn:aws:aws:controltower:ap-southeast-1::control/LYWQAOFKFCQ</td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td>arn:aws:aws:controltower:eu-central-1::control/XZSPKESDKKWI</td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td>arn:aws:aws:controltower:eu-west-1::control/ZSLSJYMDJBAL</td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td>arn:aws:aws:controltower:eu-west-2::control/OVIGFYNODQCL</td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td>arn:aws:aws:controltower:eu-north-1::control/YBHVRYAMLTFI</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td>arn:aws:aws:controltower:ap-south-1::control/TQMLIDIAQMF</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td>arn:aws:aws:controltower:ap-northeast-2::control/ZPVWRXRNPQEV</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td>arn:aws:aws:controltower:ap-northeast-1::control/KZPGJXHLYHQL</td>
<td></td>
</tr>
</tbody>
</table>
| • Europe (Paris) | | arn:aws:aws:controltower:eu-
## Control identifier

<table>
<thead>
<tr>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>west-3::control/YXSNOECRRPC0</td>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/SCWLFXFLQOLI</td>
</tr>
<tr>
<td></td>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/TGJEXSEQNIZVU</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/LNRLBVGINMLU</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/BXCUKDSYLOG</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/KZEMWMMUENGWQ</td>
</tr>
<tr>
<td></td>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/VJKMQYKKNXFI</td>
</tr>
<tr>
<td></td>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/WKBAYNVADIUJ</td>
</tr>
<tr>
<td></td>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/ZMSKYBINPWPQ</td>
</tr>
<tr>
<td></td>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/NBLQIJTVACH</td>
</tr>
<tr>
<td></td>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/IJRUYUBPGLLE</td>
</tr>
<tr>
<td></td>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/YFBPWSKFZXSI</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>CT.RDS.PR.2</td>
<td></td>
<td>Establish logging and monitoring</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CT.RDS.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.RDS.PR.2</td>
<td></td>
<td>Establish logging and monitoring</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/MQISZILRT5FB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/UPOYGSXRLUXQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/SWFLJNEBRLPT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/OHBBTGTZTMHOB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/LNDHGBLTEXAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/KOSPQCBQOVFN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/PRGUXERBXDAQ</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/YVRQVGLIXYVC</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/VVIDHBCRCKCU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/TKPUTYVQYIJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/YEQRAFYJPWA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/TCIGIZXWOEHQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/GNYSHEHSCDNB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/HRRACFGWYIYQK</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/TRBXMZVQTAQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/ELDGFUEABECL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/EENMwZAMRYD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/SPKTRDUPPMSL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td>northeast-3::control/ZNQTIVZBEPPV</td>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/DJVNWBXZYBEQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/SIATJXQTJBLB</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/PKEZXVUBPQNX</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/JWACQDFHVGTZ</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eucentral-2::control/KIUDKBGCBCKH</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.5.6</td>
<td>• Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:apsouth-2::control/XROOMLOUYAFE</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td>• Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/UTWCWVNIIJWHV</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td>• Asia Pacific (Melbourne)</td>
<td>arn:aws:controltower:apsoutheast-4::control/BKSOTGWHUEPF</td>
<td></td>
</tr>
</tbody>
</table>
## CT.RDS.PR.20

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.RDS.PR.20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Limit network access</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/MDWFCNUYEZKH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/HRHVMSEIAPUA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/DCGLYOBFDPL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/RZBFITP3J0GCG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/RKTXRGUXFWXY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/QKDQFMMGVHSO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/NLIFRFKTPNGZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/SMMMDVYGDMPEN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/JCFELHAYCQWT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/FGJJSHEZB2ZCC</td>
</tr>
</tbody>
</table>

- PCI DSS version 3.2.1
  - 8.2.1

- NIST 800-53 Rev 5
  - AC-4
  - AC-4(21)
  - SC-7
  - SC-7(11)
  - SC-7(16)
  - SC-7(21)
  - SC-7(4)
  - SC-7(5)

- PCI DSS version 3.2.1
  - 1.2.1
  - 1.3.1
  - 1.3.2
  - 1.3.4
  - 2.2
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/PJNMBTVRXJJA</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/JLKKRIKEPPCY</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/UFTOXCQOXJTV</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/QOQDDPHNUIBX</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/PYAFZHUXJWD</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/JPCMKZJLBDQX</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/KTYTTOXYIMZ</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/AZTJPXQJOFEO</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/CBIDNXQGCCKR</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/XISVVPFGPUOP</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/ZNDMUXUOQXZ</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.RDS.PR.21      |           | Protect configurations | • NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-2  
• PCI DSS version 3.2.1 2.1  
• US East (N. Virginia)  
  arn:aws:controltower:us-east-1::control/EITXJLKFTQDP  
• US East (Ohio)  
  arn:aws:controltower:us-east-2::control/IPSTNJIEWMCJ  
• US West (Oregon)  
  arn:aws:controltower:us-west-2::control/VPLSUHTSSKVN  
• Canada (Central)  
  arn:aws:controltower:ca-  
• Middle East (UAE)  
  arn:aws:controltower:me-central-1::control/UJIXWOPDCJHE  
• Asia Pacific (Melbourne)  
  arn:aws:controltower:ap-southeast-4::control/BHRDIKZLQNSU  
<p>|</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| central-1::control/ | EHEHUGUMGNMM | | • Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/GOGYCLISXJIE |
| | Asia Pacific (Singapore) | | • Europe (Frankfurt)  
ar:n:aws:controltower:eu-central-1::control/AMPSQVWJUTPB |
| | Europe (Ireland) | | • Europe (London)  
ar:n:aws:controltower:eu-west-1::control/RORRJYOCKMFU |
| | Europe (Stockholm) | | • Asia Pacific (Mumbai)  
ar:n:aws:controltower:ap-south-1::control/AONMPRGSRMUB |
| | Asia Pacific (Seoul) | | • Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-northeast-2::control/TJWPIJOAJCUN |
| | Asia Pacific (Tokyo) | | • Europe (Paris)  
ar:n:aws:controltower:eu-west-3::control/RUOZMJSADVLM |
| | South America (São Paulo) | | • US West (N. California)  
ar:n:aws:controltower:saeast-1::control/ANABXNFAQOIG |

2044
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/DHSGTIVFXFLI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/CFRTFTJQZPRW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/HFOJUBMOZXB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eusouth-1::control/JWRULGYQLUL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:afsouth-1::control/PLHGNFBDWEBP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/RFGRMBOQGIJA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/KABLIPBRIQMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eucentral-2::control/GFJUGIOTNUX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eusouth-2::control/CSKCEKSCEMJZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:apsouth-2::control/HPXNRRQXNXI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/TRHZBDVQRBFCC</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.RDS.PR.22       | • NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-2  
• PCI DSS version 3.2.1 2.1 | • Protect configurations | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/RPMQTPXETFVHVA  
• US East (Ohio) arn:aws:controltower:us-east-2::control/SUICXVBFWMTF  
• US West (Oregon) arn:aws:controltower:us-west-2::control/IGKMLWYBNQPL  
• Canada (Central) arn:aws:controltower:ca-central-1::control/NIAZABCXSRUW  
• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/SIIJEYKICNYSNA  
• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/QQQXZXYQVNBQCA  
• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/EEZQXHKPSYG  
• Europe (Ireland) arn:aws:controltower:eu-west-1::control/VCBQYURVMJWL  
• Europe (London) arn:aws:controltower:eu-west-2::control/LUZQXFAHJWGPZ |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/</td>
<td>WCAPGBYZBASC</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/</td>
<td>HYNXJZM0FCXV</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
<td>BEJUKSCJOYQQ</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/</td>
<td>OPPFSUXBZGNN</td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/</td>
<td>UNHYCCAWPERQ</td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/</td>
<td>BOBJFVMQPNFA</td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/</td>
<td>ZGRERLGFHIXR</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/</td>
<td>PNYCUZBPTAYS</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
<td>YZCWMNAUEXZ</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
<td>SWNLNNMLXINQ</td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/</td>
<td>UGDMTHYBNQGN</td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.RDS.PR.23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-21</td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/PWBOAFLALALP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/OULNFNUJHYOY</td>
</tr>
</tbody>
</table>
|                    | NIST 800-53 Rev 5 AC-3(7)|                  | US West (Oregon) arn:aws:controltower:us-
|                    | NIST 800-53 Rev 5 AC-4  |                  |                                   |
|                    | Middle East (Bahrain)    |                  |                                   |
|                    | Israel (Tel Aviv)        |                  |                                   |
|                    | Europe (Zurich)          |                  |                                   |
|                    | Asia Pacific (Hyderabad) |                  |                                   |
|                    | Middle East (UAE)        |                  |                                   |
|                    | Asia Pacific (Melbourne) |                  |                                   |

CT.RDS.PR.23
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td></td>
<td>west-2::control/BSINZXEGCDDE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* NIST 800-53 Rev 5
  • AC-4(21)
  • AC-6
  • SC-7
  • SC-7(11)
  • SC-7(16)
  • SC-7(20)
  • SC-7(21)
  • SC-7(3)
  • SC-7(4)
  • SC-7(9)

* PCI DSS version 3.2.1
  • 1.2.1
  • 1.3
  • 1.3.1
  • 1.3.2
  • 1.3.4
  • 1.3.6
  • 2.2.2

* Canada (Central)
  arn:aws:controltower:ca-central-1::control/SMCLOXFTGGJR
* Asia Pacific (Sydney)
  arn:aws:controltower:ap-southeast-2::control/ODZTKLOPEVYA
* Asia Pacific (Singapore)
  arn:aws:controltower:ap-southeast-1::control/VDMHBDVTTKRC
* Europe (Frankfurt)
  arn:aws:controltower:eu-central-1::control/DVHYJPQGNBFE
* Europe (Ireland)
  arn:aws:controltower:eu-west-1::control/TRIXKGVMMLD0
* Europe (London)
  arn:aws:controltower:eu-west-2::control/YVLLDDQUZNZT
* Europe (Stockholm)
  arn:aws:controltower:eu-north-1::control/AROTJYKQNBWL
* Asia Pacific (Mumbai)
  arn:aws:controltower:ap-south-1::control/BCJOWGCTPYMI
* Asia Pacific (Seoul)
  arn:aws:controltower:ap-northeast-2::control/NNNWMZYHHRLE
* Asia Pacific (Tokyo)
  arn:aws:controltower:ap-northeast-1::control/FTLPWMBBBBY
* Europe (Paris)
  arn:aws:controltower:eu-west-3::control/AQYGAZWDBUS
* South America (São Paulo)
  arn:aws:controltower:sa-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>east-1::control/VADBVPKYHESM</td>
<td>• US West (N. California)</td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/AMIKETYRC5QD</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/WHLMOKJDX5YGQ</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/OXCY3ZPUWJ5YP</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/AVEIAINBMSJA</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/GGFWCMFBE5G5GK</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/OJJPXBDLLIVT</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/NGM0GDLFZIX</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/FFIYPSNW5NNZ</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/FRXXRRGQQKV</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/NDCNXZRTTIH</td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/YAQQWPIJLYKN</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
</tbody>
</table>
| CT.RDS.PR.24       |           | Encrypt data at rest | • US East (N. Virginia)  
arnd:aws:controltower:us-east-1::control/  
GFMSMQJELYFTVU  
• US East (Ohio)  
arnd:aws:controltower:us-east-2::control/  
GWMSQFBQJUBL  
• US West (Oregon)  
arnd:aws:controltower:us-west-2::control/  
GWMSQFBQJUBL  
• Canada (Central)  
arnd:aws:controltower:ca-central-1::control/  
VIDEBVCPMMHE  
• Asia Pacific (Sydney)  
arnd:aws:controltower:ap-southeast-2::control/  
UGDAEBVCUTBM  
• Asia Pacific (Singapore)  
arnd:aws:controltower:ap-southeast-1::control/  
SUXNKMQDMQML  
• Europe (Frankfurt)  
arnd:aws:controltower:eu-central-1::control/  
ULFMTWYBJBH  
• Europe (Ireland)  
arnd:aws:controltower:eu-west-1::control/  
KCHAUDQMSCNA  
• CIS AWS Benchmark 1.4 2.3.1  
• NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-3(6)  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-28  
• NIST 800-53 Rev 5 SC-28(1)  
• NIST 800-53 Rev 5 SI-7(10)  
• NIST 800-53 Rev 5 SI-7(6)  
• PCI DSS version 3.2.1  
3.4  
• PCI DSS version 3.2.1  
8.2.1  
• Middle East (UAE)  
arnd:aws:controltower:me-central-1::control/  
WQOLEVYCLUDN  
• Asia Pacific (Melbourne)  
arnd:aws:controltower:ap-southeast-4::control/  
COUBETKLEMEDR  
<p>| | | | |
|                     |           |                  |                                  |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/DELGUCWUVHUC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/JZPWXKOZQLJD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/VXFCDUUYXIEX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/SBVCZOZVTNSM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/HNUUDLAMFESZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/ZIGYAICEYDYE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/MZOWKBNBAGWMM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/OZIQMHPFNUDD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/NDNGNJIYVJRO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/OTPOYEIELITW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/GCOSKKLRMNIA</td>
</tr>
</tbody>
</table>
|                    |           |                   | • Europe (Milan) arn:aws:controltower:eu-
### Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
CT.RDS.PR.25 | • NIST 800-53 Rev 5 AC-2(4)  
• NIST 800-53 Rev 5 AC-4(26) | • Establish logging and monitoring | • US East (N. Virginia)  
ar:n:ams:controltower:us-east-1::control/  
JURZAEZNDKU  
• US East (Ohio)  
ar:n:ams:controltower:us-east-1::control/  
YPTCDXVCWKKT
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(9)</td>
<td>east-2::control/MAIRETSDAVZB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td>arn:aws:controltower:us-west-2::control/MMEIVHRLITA0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td>arn:aws:controltower:ca-central-1::control/LOBRESBCCMG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/BLFCABKJKCSA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/YZYQQLIOHQYM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/LBAOBBKAGHAF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/JGEKIJTIJOSK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ZWYUQWFMKUZL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/GOGLMDRACQQC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/IZGPUUBATWSO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/JBPGLHOLPQAK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/RTJUOJNOFCR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Europe (Paris) | arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td>west-e-3::control/YRAYSIFGCCEX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/AQDTQRTKHKAIVWU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/OYUJWHZRFZAFZ</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/GAGQXONTSU5O</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/RJUT00ZNCKEA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/PUEKRODEBBUP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Milan) arn:aws:controltower:eu-south-1::control/UWTYXFPQYYBR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Africa (Cape Town) arn:aws:controltower:af-south-1::control/JMCDFXUFOTOUV</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Middle East (Bahrain) arn:aws:controltower:me-south-1::control/CVF1BPQFEOQP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/GJBEJJCSW6US</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Zurich) arn:aws:controltower:eu-central-2::control/VUXASYJIRGW</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe (Spain) arn:aws:controltower:eu-south-2::control/NOYMFJNNHNHI</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.RDS.PR.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT.RDS.PR.3</td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Improve availability</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ BOMCHCSYCWLF</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ AHXQWQFSIMC</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2(2)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ FUPEXQIQKQB</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-3</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ CKRDGKNEBYT</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ KGVGC3JNUMNIIY</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ BLX3DLBAERSW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eucentral-1::control/ GFFRRVSFMQUD</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/ N0CXMM0IKGCV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/ IHCWENDKGMLN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/ XBFXYKVR0TZV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/ VFYDRPBPNBNT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/ WECZXHVKCNL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/ ZTRXNVYWDKTO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/ XDFKDLURAACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/ OFANCHVHOJOV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/ HBGKZQQAFL0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/ MPNWLSENZAYK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/ DHDWZPYMBDIM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
</tbody>
</table>
|                    |           |                  | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.RDS.PR.4</td>
<td></td>
<td></td>
<td>• Use strong authentication</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BKROXVTVPME</td>
</tr>
<tr>
<td>CT.RDS.PR.4</td>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AC-2(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EYGXDYJGBYNQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Cape Town) arn:aws:controltower:af-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TGAKQPBVZILV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WJHSOJNUAKYU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CUMZTYFVSMQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>YDKULBFMYXBM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AQUESOKIFERI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BKROXVTVPME</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CMCSHPTEJ3DL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ZUSHTBGQWBDP</td>
</tr>
</tbody>
</table>

**CT.RDS.PR.4**
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td></td>
<td>east-1::control/QWKDSRLGQJQM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US East (Ohio)</td>
<td>• arn:aws:controltower:us-east-2::control/ERWYFDEVMQSK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td>• arn:aws:controltower:us-west-2::control/LNVGZRHRGEBU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td>• arn:aws:controltower:ca-central-1::control/OMCIOG0MPFXW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td>• arn:aws:controltower:ap-southeast-2::control/VGJUWQWNLFDY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>• arn:aws:controltower:ap-southeast-1::control/NVTXXXQXQYV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>• arn:aws:controltower:eu-central-1::control/SJFWA3ASSZNA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>• arn:aws:controltower:eu-west-1::control/UGLMTZCZLMRP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>• arn:aws:controltower:eu-west-2::control/TBMMCCZ5FKR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>• arn:aws:controltower:eu-north-1::control/HIBKFWHCHZWB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>• arn:aws:controltower:ap-south-1::control/MZCFSYUEMWCM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>• arn:aws:controltower:ap-northeast-2::control/MCP5SKM2ZQIMN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo) | • arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeast-1::control/KNMITJGFKRMB</td>
<td></td>
<td></td>
<td>• Europe (Paris) ern:aws:controltower:eu-west-3::control/NVTLRUXITEZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) ern:aws:controltower:saeast-1::control/BXTYCUJTDLQW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) ern:aws:controltower:us-west-1::control/HTMSOGEZDTJN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) ern:aws:controltower:ap-east-1::control/QNCIRBGCPRTM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) ern:aws:controltower:ap-southeast-3::control/PDCYLMOTAIMM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) ern:aws:controltower:ap-northeast-3::control/TQTYWXQTLKU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) ern:aws:controltower:eu-south-1::control/QRLJHHAXNPNO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town) ern:aws:controltower:af-south-1::control/FUZQRUKMSLX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain) ern:aws:controltower:me-south-1::control/FU0EE6GFJKSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv) ern:aws:controltower:il-central-1::control/TWMLGZJPMUQV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich) ern:aws:controltower:eu-central-2::control/VFCIZSFITDKR</td>
</tr>
</tbody>
</table>
## Control identifier: CT.RDS.PR.5

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.RDS.PR.5</td>
<td></td>
<td>Manage vulnerabilities</td>
<td><img src="" alt="API identifiers" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="" alt="API identifiers" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="" alt="API identifiers" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="" alt="API identifiers" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="" alt="API identifiers" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="" alt="API identifiers" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="" alt="API identifiers" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="" alt="API identifiers" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="" alt="API identifiers" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="" alt="API identifiers" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="" alt="API identifiers" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><img src="" alt="API identifiers" /></td>
</tr>
</tbody>
</table>

- **Europe (Spain)**
  - arn:aws:controltower:eu-south-2::control/OGXVZKWCSZQ
- **Asia Pacific (Hyderabad)**
  - arn:aws:controltower:ap-south-2::control/KROECLWFGBSA
- **Middle East (UAE)**
  - arn:aws:controltower:me-central-1::control/BIZEYWPAADCF
- **Asia Pacific (Melbourne)**
  - arn:aws:controltower:ap-southeast-4::control/PDN0CLDCOFWK

**NIST 800-53 Rev 5**
- SI-2
- SI-2(2)
- SI-2(4)
- SI-2(5)
- PCI DSS version 3.2.1 6.2

**Manage vulnerabilities**
- **US East (N. Virginia)**
  - arn:aws:controltower:us-east-1::control/UNSIZPXYZKNF
- **US East (Ohio)**
  - arn:aws:controltower:us-east-2::control/IPSVYZQYJLE
- **US West (Oregon)**
  - arn:aws:controltower:us-west-2::control/ADSPCSQCOOJB
- **Canada (Central)**
  - arn:aws:controltower:ca-central-1::control/AORICFZECLO5
- **Asia Pacific (Sydney)**
  - arn:aws:controltower:ap-southeast-2::control/VTWUXLYFANSP
- **Asia Pacific (Singapore)**
  - arn:aws:controltower:ap-southeast-1::control/VDGRXWMZIRC8
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (Frankfurt)</td>
<td>Arn:aws:controltower:eu-central-1::control/ BPSCCXVAHRQW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>Arn:aws:controltower:eu-west-1::control/ JPXAHOOKUHZP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>Arn:aws:controltower:eu-west-2::control/ DOWVUVZPLQQT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>Arn:aws:controltower:eu-north-1::control/ TBCPKYPZUHEB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>Arn:aws:controltower:ap-south-1::control/ NJPGNYPGMDZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>Arn:aws:controltower:ap-northeast-2::control/ JEXJDXDQRFV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>Arn:aws:controltower:ap-northeast-1::control/ NXXKTRZGYRZU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>Arn:aws:controltower:eu-west-3::control/ IUDQBUPZD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>Arn:aws:controltower:sa-east-1::control/ UJWRDMFPBFQR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>Arn:aws:controltower:us-west-1::control/ WRUFJIMHOAGF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>Arn:aws:controltower:ap-east-1::control/ XSRLBIOJYKOP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Asia Pacific (Jakarta) | Arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| southeast-3::control/MREYKJJSEDQR | • Asia Pacific (Osaka)  
arn:aws:controltower:ap-northeast-3::control/BEBKSCLLCLJN |  |  |
| | • Europe (Milan)  
arn:aws:controltower:eu-south-1::control/TTGYMVAMNNN |  |  |
| | • Africa (Cape Town)  
arn:aws:controltower:af-south-1::control/WHRSJIOZFXIL |  |  |
| | • Middle East (Bahrain)  
arn:aws:controltower:me-south-1::control/ODOFJMTXGTYJ |  |  |
| | • Israel (Tel Aviv)  
arn:aws:controltower:il-central-1::control/XQNBEQGNVQWD |  |  |
| | • Europe (Zurich)  
arn:aws:controltower:eu-central-2::control/FEDUDFHBPLRA |  |  |
| | • Europe (Spain)  
arn:aws:controltower:eu-south-2::control/CBPJFXGZJJNU |  |  |
| | • Asia Pacific (Hyderabad)  
arn:aws:controltower:ap-south-2::control/RMEBOYADJYHO |  |  |
| | • Middle East (UAE)  
arn:aws:controltower:me-central-1::control/BHBFDGUXCD |  |  |
| | • Asia Pacific (Melbourne)  
arn:aws:controltower:ap-southeast-4::control/UAZNUSWPRZEL |  |  |
## CT.RDS.PR.6

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.RDS.PR.6         | • NIST 800-53 Rev 5 CP-10  
                      • NIST 800-53 Rev 5 CP-6  
                      • NIST 800-53 Rev 5 CP-6(1)  
                      • NIST 800-53 Rev 5 CP-6(2)  
                      • NIST 800-53 Rev 5 CP-9  
                      • NIST 800-53 Rev 5 SI-13(5)  
                      • PCI DSS version 3.2.1  
                      3.1                   | • Improve resiliency | • US East (N. Virginia)  
                      arn:aws:controltower:us-east-1::control/HQCINNCIEGJ  
                      • US East (Ohio)  
                      arn:aws:controltower:us-east-2::control/WKLWHFDFYGDIM  
                      • US West (Oregon)  
                      arn:aws:controltower:us-west-2::control/JPFWRQLJMFBF  
                      • Canada (Central)  
                      arn:aws:controltower:ca-central-1::control/BPTUKBOEYVEC  
                      • Asia Pacific (Sydney)  
                      arn:aws:controltower:ap-southeast-2::control/MKNYWNVXKUKG  
                      • Asia Pacific (Singapore)  
                      arn:aws:controltower:ap-southeast-1::control/BKETESOGWHJT  
                      • Europe (Frankfurt)  
                      arn:aws:controltower:eu-central-1::control/YUYGWRMEUF5  
                      • Europe (Ireland)  
                      arn:aws:controltower:eu-west-1::control/LZHVTVJQMRF  
                      • Europe (London)  
                      arn:aws:controltower:eu-west-2::control/IWNFMALKADEV  
                      • Europe (Stockholm)  
                      arn:aws:controltower:eu-north-1::control/PQYEUKWXZFFK  
                      • Asia Pacific (Mumbai)  
                      arn:aws:controltower:ap-south-1::control/GFPSZSMOHKB |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/AUDENQFLMRAQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/SAJABIITSFKW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/MLKHAVAVXXZM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/SKHLGRRUWSCQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/SWMFSRJRGJPJD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/UUNYJSTQKTKJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/AWIDKJQLXFPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/THQFQNGGWMDY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/EUQFJEGDJXIV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/EHAIIRIIGWOHC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/ZIERFRYDVTUX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.RDS.PR.7         | • NIST 800-53 Rev 5 AC-2(1) | • Use strong authentication | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/EFJRJBUAWLRY  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/GPJEV0WCHSF  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/JVVEQIUBJOL  
• Canada (Central)  
arn:aws:controltower:can-central-1::control/OPZBMSCIYMQF  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-2066 |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>southeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BVMANNHHVIIJ</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>southeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GUBOIKVNFVYHI</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RVULDNZJGTET</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LHNLJBQEXQKS</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>west-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RNNKLRXFOADP</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>north-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BUGV01NBXVCD</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BFMJVHXZBFLS</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>northeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ZEUMSSLIZDTU</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>northeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EGQHMBKZVKBO</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>west-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UNIXACIXZYG5</td>
</tr>
<tr>
<td>• South America</td>
<td></td>
<td></td>
<td>(São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>YPOIUGJSDDKUC</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td>California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PHVHLAPJJQGY</td>
</tr>
</tbody>
</table>

2067
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/DQCWDYIPYKCU</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/VBTYNZMJDITYX</td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/RXRBKIOMOQMD</td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/YVBIFQKSPZS</td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/PPZASLHINUOG</td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/JOFZXRMJAVNR</td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/RPAEKITMRPNP</td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/KCONVMJASONKK</td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/CXYFUXAJHPQ8</td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/MXFSCTLATSLV</td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/XXPPJUVQZLJF</td>
</tr>
</tbody>
</table>
| Asia Pacific (Melbourne) |  | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.RDS.PR.8 | • NIST 800-53 Rev 5 CP-10  
• NIST 800-53 Rev 5 CP-6  
• NIST 800-53 Rev 5 CP-6(1)  
• NIST 800-53 Rev 5 CP-6(2)  
• NIST 800-53 Rev 5 CP-9  
• NIST 800-53 Rev 5 SC-5(2)  
• NIST 800-53 Rev 5 SI-12  
• NIST 800-53 Rev 5 SI-13(5)  
• PCI DSS version 3.2.1 10.7  
• PCI DSS version 3.2.1 3.1 | • Improve resiliency | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/QETNOOVEKOPL  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/ZFQDCNLBJOYT  
• US West (Oregon)  
arn:aws:aws:controltower:us-west-2::control/DJXFRUONYPS5  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/LOKAOGKSIRIC  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/GJKUHDAQKCNH  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/COJXSEKFZEKX  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/XJJUSTKGBGZ  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/RKNNKLPLALQZ  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/XXUEQVOEMTBR  
• Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/DTQ6IURTBPQD |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/UZSCDCCOFFDX</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/HODAQWGBTRMP</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/DOHISTPWUUOA</td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ CUKMUUVZEVMY</td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/GMWCTVXRUZFJ</td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/DHYENBVXUTBF</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/IMAZHUOGWZDJ</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/UKEUAXDWWKOV</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/AZHRBFNVKFN0</td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/YBNFFKWIEBFT</td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/HBYTLQZEWBJU</td>
<td></td>
</tr>
</tbody>
</table>
| • Middle East (Bahrain) | arn:aws:controltower:me-
### CT.RDS.PR.9

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.RDS.PR.9</td>
<td></td>
<td>Protect configurations</td>
<td>• Use: aws:controltower:us-east-1::control/RDSGPIJXTCPR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Use: aws:controltower:us-east-2::control/LUAEEZMOLTOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Use: aws:controltower:us-west-2::control/QQXOAGHMUVGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Use: aws:controltower:ca-2071::control/</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>central-1::control/ RGEVPCAEAGOUT</td>
<td>Asia Pacific (Sydney)</td>
<td>Control tower API identifiers</td>
<td>arn:aws:controltower:ap-southeast-2::control/ TBPEWMYHECGP</td>
</tr>
<tr>
<td>central-1::control/ YKDWRIDUGSBS</td>
<td>Asia Pacific (Singapore)</td>
<td>Control tower API identifiers</td>
<td>arn:aws:controltower:ap-southeast-1::control/ YKDWRIDUGSBS</td>
</tr>
<tr>
<td>central-1::control/ NCDQF5WPREWO</td>
<td>Europe (Frankfurt)</td>
<td>Control tower API identifiers</td>
<td>arn:aws:controltower:eu-central-1::control/ NCDQF5WPREWO</td>
</tr>
<tr>
<td>central-1::control/ MHMKSJQAYVLS</td>
<td>Europe (Ireland)</td>
<td>Control tower API identifiers</td>
<td>arn:aws:controltower:eu-west-1::control/ MHMKSJQAYVLS</td>
</tr>
<tr>
<td>central-1::control/ JQYWTSKVECRE</td>
<td>Europe (London)</td>
<td>Control tower API identifiers</td>
<td>arn:aws:controltower:eu-west-2::control/ JQYWTSKVECRE</td>
</tr>
<tr>
<td>central-1::control/ UVODEUMUUGCL</td>
<td>Europe (Stockholm)</td>
<td>Control tower API identifiers</td>
<td>arn:aws:controltower:eu-north-1::control/ UVODEUMUUGCL</td>
</tr>
<tr>
<td>central-1::control/ LWNVYRAQUEITA</td>
<td>Asia Pacific (Mumbai)</td>
<td>Control tower API identifiers</td>
<td>arn:aws:controltower:ap-south-1::control/ LWNVYRAQUEITA</td>
</tr>
<tr>
<td>central-1::control/ PPBGZVGLKLZM</td>
<td>Asia Pacific (Seoul)</td>
<td>Control tower API identifiers</td>
<td>arn:aws:controltower:ap-northeast-2::control/ PPBGZVGLKLZM</td>
</tr>
<tr>
<td>central-1::control/ DUHRNEVTDQKM</td>
<td>Asia Pacific (Tokyo)</td>
<td>Control tower API identifiers</td>
<td>arn:aws:controltower:ap-northeast-1::control/ DUHRNEVTDQKM</td>
</tr>
<tr>
<td>central-1::control/ RWJVBPHGKPEJ</td>
<td>Europe (Paris)</td>
<td>Control tower API identifiers</td>
<td>arn:aws:controltower:eu-west-3::control/ RWJVBPHGKPEJ</td>
</tr>
<tr>
<td>central-1::control/ UPFHHERCEHQ</td>
<td>South America (São Paulo)</td>
<td>Control tower API identifiers</td>
<td>arn:aws:controltower:sa-east-1::control/ UPFHHERCEHQ</td>
</tr>
<tr>
<td>central-1::control/</td>
<td>US West (N. California)</td>
<td>Control tower API identifiers</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/</td>
<td>DUDBSXSHNGZK</td>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/XOJJRCGDSROH</td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/WXVMXZMJQSE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/NHSQTNZHEZZM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/BDWHSYNRGSRD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/XNLOIHBXMWEG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/KXFJPHMTSFCI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/ULJWZCETBUSG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/RPARTQJEXFFY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/UNAQZKVPTXJA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/CBIQOKTFVUYG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/UUCKIWMNBHJH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/HOLURSRFKYOJ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/GAHFLNXXDQAI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/XOLIHOFJYQQAC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/XWIEKIDYHHHM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/YSSTTBPHVAIZB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/TPICKZDHGROC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/VLKCIDKWSEG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/KTGZUAACWWFG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/CXPPSTSYQSR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/NVNGHXGLPTOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/OPDYRIMRTAUP</td>
</tr>
</tbody>
</table>
| | | | • Africa (Cape Town) arn:aws:controltower:af-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.REDSHIFT.PR.2</td>
<td></td>
<td>Improve resiliency</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/VVOXNJWVCUTCZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/LGKVXBCGWHZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/THILQHSAMURX</td>
</tr>
</tbody>
</table>

**CT.REDSHIFT.PR.2**

- **NIST 800-53 Rev 5 CP-10**
- **NIST 800-53 Rev 5 CP-6**
- **NIST 800-53 Rev 5 CP-6(1)**
- **NIST 800-53 Rev 5 CP-6(2)**
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 CP-9</td>
<td></td>
<td></td>
<td>west-2::control/JWHYTRNOCADA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/CXVHRDCRTVRV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/CMTVPJTIHQTJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/KGGVMOPMMNLR</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.1</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/CDAXWWXQMPDQ</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/XYBNZEOWJKVT</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/YYWXMDUAJRMJ</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/WLAMYNHXYVHV</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/CUMXJYSQUGQJ</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ZQKOUCATIZVK</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/VJJIDPSRYYRH</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/BMVHVJQCPZYCS</td>
</tr>
</tbody>
</table>
| • | | | South America (São Paulo) arn:aws:controltower:sa-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>TABJCEBJRTAX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/ AADIBNWLNLVR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/ Q2YFQVJMMFFG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/ IVYOUWQFCUYQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/ AWOJXMBCSBDF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/ ESNVKLAVFXDT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/ AIZBOYVPCGKE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/ IBCJZFVMRGAI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/ PWAHNLYTVHXP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/ UKMXNAGLWZPR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/ JREFGBNJXLVP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/ NQJJKYHZHLFG</td>
</tr>
</tbody>
</table>
### CT.REDSHIFT.PR.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.REDSHIFT.PR.3   |           | • Establish logging and monitoring | • Middle East (UAE)  
arn:aws:controltower:me-central-1::control/  
NWNIKXIDOVHQ  
• Asia Pacific  
(Melbourne)  
arn:aws:controltower:ap-southeast-4::control/  
MZVFTANPECPP |
|                    | NIST 800-53 Rev 5 AC-2(4) |                          | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
TIGRGIIIZLNY |
|                    | NIST 800-53 Rev 5 AC-4(26) |                          | • US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
BUXRPTPENLDJ |
|                    | NIST 800-53 Rev 5 AC-6(9) |                          | • US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
MPMZYZBUMBNY |
|                    | NIST 800-53 Rev 5 AU-10 |                          | • Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
XBMWWNUQRGVG |
|                    | NIST 800-53 Rev 5 AU-12 |                          | • Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
TZMQFTLFIYII |
|                    | NIST 800-53 Rev 5 AU-2 |                          | • Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
NHBUDLMZCDGY |
|                    | NIST 800-53 Rev 5 AU-3 |                          | • Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/  
DZIEVGPLAKY |
|                    | NIST 800-53 Rev 5 AU-6(3) |                          | • Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/  
YTMBWVLJKBOM |
|                    | NIST 800-53 Rev 5 AU-6(4) |                          |                             |
|                    | NIST 800-53 Rev 5 CA-7 |                          |                             |
|                    | NIST 800-53 Rev 5 SC-7(9) |                          |                             |
|                    | NIST 800-53 Rev 5 SI-3(8) |                          |                             |
|                    | NIST 800-53 Rev 5 SI-4(20) |                          |                             |
|                    | NIST 800-53 Rev 5 SI-7(8) |                          |                             |
|                    | PCI DSS version 3.2.1  
10.1 |                          |                             |
|                    | PCI DSS version 3.2.1  
10.2.1 |                          |                             |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 10.2.2</td>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ BYITFEKIBLPM</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ BYITFEKIBLPM</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ LLXYEQQOHCLQ</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ UJYYBSZJHUBP</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ UJYYBSZJHUBP</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ OCMBDMYSNHXV</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ JTZSPKUCZEPQ</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ OFQPEFLHYEMQ</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ FCEWLKLDUMMV</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ LTBATEGEWNMS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ XNBKEABVQQA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ SXXAL100DJEC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------</td>
<td>----------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.4</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Manage vulnerabilities</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/RSYQCHSJXMVQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-1::control/RSYQCHSJXMVQ</td>
</tr>
</tbody>
</table>

CT.REDSHIFT.PR.4

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.REDSHIFT.PR.4</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Manage vulnerabilities</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/RSYQCHSJXMVQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-1::control/RSYQCHSJXMVQ</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-9</td>
<td></td>
<td></td>
<td>east-2::control/XXNRIFUPYIEH</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ TIIO2LVLHDXUZ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ NHAZVUYWKCQG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(2)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ DJVYVYTHHNU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(4)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ CVUXESSIFPVI</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(5)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ ZZCS5SWYMRZNT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 6.2</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ DZFJXHGW explosives</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ KZSEI0DFFYYQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ NHBHTNDISQFC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ JTYKTCMBNI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ PLW5DPNQNCXK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ SHAQANKLSRUD</td>
</tr>
</tbody>
</table>
| | | | Europe (Paris) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>west-3::control/EAQFE0AZNUWH</td>
<td>• South America (São Paulo)</td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/RYCBSKQC5OSH</td>
</tr>
<tr>
<td>south-3::control/MITJG0N6LNY</td>
<td>• US West (N. California)</td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/FLUEZVKKETL</td>
</tr>
<tr>
<td>south-3::control/QY63ZT0AFNX</td>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/MKNPMKKEUYRW</td>
</tr>
<tr>
<td>south-3::control/O6L529463YK</td>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/OCSPCZDHKDRV</td>
</tr>
<tr>
<td>south-3::control/R7QY1H668ER</td>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/DDJAP0KCNVKJ</td>
</tr>
<tr>
<td>south-3::control/K6838339PRW</td>
<td>• Europe (Milan)</td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/QZFMVCTTFKUN</td>
</tr>
<tr>
<td>south-3::control/G2Q6T7N968R</td>
<td>• Africa (Cape Town)</td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/VIBECVZYWWSO</td>
</tr>
<tr>
<td>south-3::control/I9839Q8236R</td>
<td>• Middle East (Bahrain)</td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/SQIZKBESVMUI</td>
</tr>
<tr>
<td>south-3::control/79F298Q349R</td>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/SKQGXVYSUDGZ</td>
</tr>
<tr>
<td>south-3::control/74839Q2843R</td>
<td>• Europe (Zurich)</td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/IGBVEZKDDPSK</td>
</tr>
<tr>
<td>south-3::control/29Q3879438R</td>
<td>• Europe (Spain)</td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/UJAYQIQWPSSB</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/GOOMD2FIUXFU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/MTESQPEDNIXA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/IKCXEHXYJRVE</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.5</td>
<td></td>
<td></td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/MVWDSPIZQQOJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/GNIRCUDXTGIB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/KXJMDJMVXXWA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/JNUTH1PQTCKAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/ICXWYEEBLZWH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/VEV0VF2BXPGU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/XZUANQWNMUNB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.REDSHIFT.PR.5</td>
<td></td>
<td></td>
<td>• Limit network access</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/MVWDSPIZQQOJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/GNIRCUDXTGIB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/KXJMDJMVXXWA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/JNUTH1PQTCKAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/ICXWYEEBLZWH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/VEV0VF2BXPGU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/XZUANQWNMUNB</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/BVBKQCZDVESL</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/JDKXPTYUTQNJ</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/TZHUCKEGTHPU</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/JVRPBPFUFUXUO</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/DMBWTCJTWZZK</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/DWDPLRTCTQRK</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/OKSRLWUXPPFR</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/RSKDADCDBGTGR</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/OUGVPYTMYFZA</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/MRBTPRXKSMJJ</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/TAQQKKFUBKOK</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Osaka) |           |                  | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>

- **Control identifier**: CT.REDSHIFT.PR.6
- **Framework**: NIST 800-53 Rev 5 CA-9(1)
- **Control objective**: Protect configurations
- **Control API identifiers, by Region**: US East (N. Virginia) arn:aws:controltower:us-2086
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| • NIST 800-53 Rev 5 CM-2  
• PCI DSS version 3.2.1 2.1 | | | east-1::control/  
XSJE0QVHQNNS  
US East (Ohio)  
ar:n:aws:controltower:us-east-1::control/  
QLHEQGBWIVX  
US West (Oregon)  
ar:n:aws:controltower:us-west-1::control/  
ESTXKKJ9OWTE  
Canada (Central)  
ar:n:aws:controltower:ca-central-1::control/  
LXRTGBKAVZRM  
Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/  
KCSAGKLDYNMB  
Asia Pacific (Singapore)  
ar:n:aws:controltower:ap-southeast-1::control/  
OWUUGCUFZT7V  
Europe (Frankfurt)  
ar:n:aws:controltower:eu-central-1::control/  
IZQOHUIEKSPL  
Europe (Ireland)  
ar:n:aws:controltower:eu-west-1::control/  
RQFUXCKKBOY  
Europe (London)  
ar:n:aws:controltower:eu-west-2::control/  
REMWHXAQQGLG  
Europe (Stockholm)  
ar:n:aws:controltower:eu-north-1::control/  
GGTQQVUSX00X  
Asia Pacific (Mumbai)  
ar:n:aws:controltower:ap-south-1::control/  
IQZBJ2UENGUEF  
Asia Pacific (Seoul)  
ar:n:aws:controltower:ap-northeast-2::control/  
CYDXYNPZIPG  
Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeeast-1::control/KWCQMVWGTOPI</td>
<td>• Europe (Paris)</td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/YGXMJYPAWNUM</td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo)</td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/BBVVBFXFHERG</td>
</tr>
<tr>
<td></td>
<td>• US West (N. California)</td>
<td></td>
<td>arn:aws:aws:controltower:us-west-1::control/GUDFCNXFHQRC</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/EPJUXEYJYXRT</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/DUBPDVDVASSS</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/XKEHDXOAJSOF</td>
</tr>
<tr>
<td></td>
<td>• Europe (Milan)</td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/IOUDDYHBUYE</td>
</tr>
<tr>
<td></td>
<td>• Africa (Cape Town)</td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/KSKJRCBRSNUY</td>
</tr>
<tr>
<td></td>
<td>• Middle East (Bahrain)</td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/ITWHNJQRRKHM</td>
</tr>
<tr>
<td></td>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/NFCFLXUIDXUY</td>
</tr>
<tr>
<td></td>
<td>• Europe (Zurich)</td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/UDPJFGOSU0GB</td>
</tr>
</tbody>
</table>

2088
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.REDSHIFT.PR.7</td>
<td></td>
<td>Protect</td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>configurations</td>
<td>arn:aws:controltower:eu-south-2::control/VKKSW5ROCOHY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/PQYAZYVNWBKB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/TKWVUVUVFRVV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-4::control/BGMYCCKJSXSJ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.REDSHIFT.PR.7</td>
<td></td>
<td>Protect</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>configurations</td>
<td>arn:aws:controltower:us-east-1::control/XTYTGDHUVKCR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/KEUUEIKIJOQLO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/QIVJI3P0KQWA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/QJZMC3VYC0GM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/IJMLIAVPGI0I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/FFXXGIYUCWRF</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/XDROKAR002ZXXV</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/BDFXKXOPITMD</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/QHWQGOVEKIMH</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/LHKEIALJJGLM</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/WPOFSGASMAFH</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/IDYGEAGGLAKL</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/VURRKCWICXXJ</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/QPDQNIGEIGEQ</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/GOVQURJVCYFX</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/NHUXUVVUUYB</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/FQVGIHDKJGRS</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------</td>
<td>-------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>southeeast-3::control/ MGZSFFRPJXNJ</td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/ QPKHXJWAXBEC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/ ZSFLWFHDEDNW</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/ PHPNUHCYJIVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/ TYRAJFRQSVFZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/ VLLTDCCBFDKE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/ RLOOAQWDOHBT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/ PVWOHRPFVQOI</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/ MBZBYWQHYYAYA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/ XIOAMLVHEFKA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/ HDAJ0FBSCAZC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## CT.REDSHIFT.PR.8

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.REDSHIFT.PR.8   | • NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-3(6)  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-28  
• NIST 800-53 Rev 5 SC-28(1)  
• NIST 800-53 Rev 5 SI-7(6)  
• PCI DSS version 3.2.1  
• PCI DSS version 3.2.1 3.4  
• PCI DSS version 3.2.1 3.5.3  
• PCI DSS version 3.2.1 8.2.1 | Encrypt data at rest | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/EXRBSBFPHWH  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/ALDUSRLWSQYM  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/WIYETXTZFNNA  
• Canada (Central)  
ar:n:aws:controltower:ca-central-1::control/XQGTRHZSJLAS  
• Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/KLGGBJEKOFDX  
• Asia Pacific (Singapore)  
ar:n:aws:controltower:ap-southeast-1::control/VUQFIBCPLZJU  
• Europe (Frankfurt)  
ar:n:aws:controltower:eu-central-1::control/EDEMXYSBQDNA  
• Europe (Ireland)  
ar:n:aws:controltower:eu-west-1::control/DXPHOEZMTQCV  
• Europe (London)  
ar:n:aws:controltower:eu-west-2::control/LQVZJBEQJCJF  
• Europe (Stockholm)  
ar:n:aws:controltower:eu-north-1::control/TNFUUTWWQVS  
• Asia Pacific (Mumbai)  
ar:n:aws:controltower:ap-south-1::control/WIWQCADWSSNJ |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/DHAUGOPSEXMD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/HIUBOYXXZOJV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/KCHBFRLRIUXU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/WUMVGWBHXOSQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/BMSSGAJEZHXB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/GZGTVBEKIPRM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/LLUVKHVOTYKB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/EZYWKAOZZJCS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/TJTGHSRKCAD0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/NHQCIPXIUDQD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/GACOYCVFQSZR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.REDSHIFT.PR.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>Encrypt data in transit</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ YRDLVNFRHKTV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ AOSGZRUCDETW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ UHRGJXFBHKWG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ JTOTPCSGXPIF</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>southeast-2::control/QUEUIRFGEGYFM</td>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/VLSFPDQGTLEZ</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/KRQCDKYDLPLN</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:aws:controltower:eu-west-1::control/ARYLQ5RCCCSQ</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:aws:controltower:eu-west-2::control/SOXMPAPMUVYA</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:aws:controltower:eu-north-1::control/CHKKYPAFD1GQ</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:aws:controltower:ap-south-1::control/VAKAVHYBZRLT</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:aws:controltower:ap-northeast-2::control/FGEPJZCARZGW</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:aws:controltower:ap-northeast-1::control/NSOMCBUHYDTR</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:aws:controltower:eu-west-3::control/MPEDXGXIBIJD</td>
</tr>
<tr>
<td>• South America (Sào Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:aws:controltower:sao-east-1::control/BET2XWFTG5S</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:aws:controltower:us-west-1::control/FPGMP3BMHZJM</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/QWLIPUMZTIWBI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/BEDTSJXBPYRQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/WMHZVKNJBHHH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/ERJCSG1BWXPXF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/DJABACHZCYDN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/CKWTJ0GCUKLB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/XIYTJTWXILFJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/KCIJAYENPABP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/KZYEYENTRZSQB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/NXYUJNVDTBZP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/ZVIYJKXRKFLQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Melbourne)</td>
<td>arn:aws:controltower:ap-2096</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## CT.S3.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.S3.PR.1         | • CIS AWS Benchmark 1.4 2.1.3  
                        • CIS AWS Benchmark 1.4 2.1.5  
                        • NIST 800-53 Rev 5 AC-21  
                        • NIST 800-53 Rev 5 AC-3  
                        • NIST 800-53 Rev 5 AC-3(7)  
                        • NIST 800-53 Rev 5 AC-4  
                        • NIST 800-53 Rev 5 AC-4(21)  
                        • NIST 800-53 Rev 5 AC-6  
                        • NIST 800-53 Rev 5 AU-9  
                        • NIST 800-53 Rev 5 AU-9(2)  
                        • NIST 800-53 Rev 5 CA-9(1)  
                        • NIST 800-53 Rev 5 CM-3(6)  
                        • NIST 800-53 Rev 5 CP-10  
                        • NIST 800-53 Rev 5 CP-6  
                        • NIST 800-53 Rev 5 CP-6(1)  
                        • NIST 800-53 Rev 5 CP-6(2)  
                        • NIST 800-53 Rev 5 CP-9  
                        • NIST 800-53 Rev 5 SC-12(2)  
                        • NIST 800-53 Rev 5 SC-13 | • Enforce least privilege | }
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Asia Pacific (Mumbai)**
  arn:aws:controltower:ap-south-1::control/DVUQCYQSSVYJ
- **Asia Pacific (Seoul)**
  arn:aws:controltower:ap-northeast-2::control/YKBYYOORVYDQ
- **Asia Pacific (Tokyo)**
  arn:aws:controltower:ap-northeast-1::control/HCYQILOQAJGP
- **Europe (Paris)**
  arn:aws:controltower:eu-west-3::control/LHQHRBBJLOGX
- **South America (São Paulo)**
  arn:aws:controltower:sa-east-1::control/GZSJOJMETOS
- **US West (N. California)**
  arn:aws:controltower:us-west-1::control/UQDZOTYXHSTJV
- **Asia Pacific (Hong Kong)**
  arn:aws:controltower:ap-east-1::control/HUGVRCBOTHNS
- **Asia Pacific (Jakarta)**
  arn:aws:controltower:ap-southeast-3::control/RDNVZK0XQRBS
- **Asia Pacific (Osaka)**
  arn:aws:controltower:ap-northeast-3::control/BTKLXIPPGWE
- **Europe (Milan)**
  arn:aws:controltower:eu-south-1::control/FAKNTWCVJBBQ
- **Africa (Cape Town)**
  arn:aws:controltower:af-south-1::control/LRNINXJVLTOI
- **Middle East (Bahrain)**
  arn:aws:controltower:me-
### Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
CT.S3.PR.10 | • NIST 800-53 Rev 5 AU-9  
• NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-3(6)  
• NIST 800-53 Rev 5 SC-12(2)  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-28  
• NIST 800-53 Rev 5 SC-28(1) | • Encrypt data at rest | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/AZDYETJYRMPN  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/WOZMw2TCVLIG  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/NAJMVANQPXLA  
• Canada (Central)  
arn:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td>central-1::control/ KLKYRVHDTSNQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/XUAWJGYLSKOU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/XOELW6G1RTRJP</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.2</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/HWUKCNHPQJPH</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/WHMUOFUNYXLE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.5.3</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/YJTF5SUPUCWSG</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/IXTXSHOTVUMZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ZVWFN8TVBJQX</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/AJJZIPUOMHEPW</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/SDKAABURVROM</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/LOBZ2YSSXKUEK</td>
</tr>
<tr>
<td>• South America</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/PJFOMSANDLYK</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/QVGKLBYATRGP</td>
<td>Longitude 113.33 °E</td>
<td>Global</td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/UNAIRGRMSYDE</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/SZTUWIPCDHJA</td>
<td>Longitude 106.82 °E</td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/IGACBRUIDHR</td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/TRPKH1GMYXAO</td>
<td>Longitude 9.17 °E</td>
<td>• Europe (Cape Town) arn:aws:controltower:af-south-1::control/NPHPUZYHVOIY</td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/EXSNHHJHWXIO</td>
<td>Longitude 24.58 °E</td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/RZGKGPDPNNWZ</td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/ZQTVAGIGWJJ</td>
<td>Longitude 9.17 °E</td>
<td>• Europe (Spain) arn:aws:controltower:eu-south-2::control/SKGFYRDINXPI</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/HWTOUVQXQKRI</td>
<td>Longitude 72.93 °E</td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/QZDCTXC3VTPY</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.S3.PR.11        | CIS AWS Benchmark 1.4 2.1.3  
• NIST 800-53 Rev 5 AU-9(2)  
• NIST 800-53 Rev 5 CP-10  
• NIST 800-53 Rev 5 CP-6  
• NIST 800-53 Rev 5 CP-6(1)  
• NIST 800-53 Rev 5 CP-6(2)  
• NIST 800-53 Rev 5 CP-9  
• NIST 800-53 Rev 5 SC-5(2)  
• NIST 800-53 Rev 5 SI-12  
• NIST 800-53 Rev 5 SI-13(5)  
• PCI DSS version 3.2.1 10.5.5 | • Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
EUDRJBCNBNX  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
QTKNBWIZPYQ  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
YMXFKEAZKVCSS  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
CRTXZSITMULU  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
EUTFKNVBWID  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
NFQVRNQZULR  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/  
KYZJOMCMHGS  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/  
OUGCAQGTCFFY  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/  
IHLVKAQRYKMH  
• Asia Pacific (Melbourne)  
arn:aws:controltower:ap-southeast-4::control/  
UZNLVDVSMMHC |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/DEHWLVENFBQF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/YRTTYKIGHXEJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/CCSEYYSAWFKKX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/AQQOPEFIXRLY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/LGLAYYE0E0EN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/QMKXNBUDMJAF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/UHIVEVHOTRUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/VYSILMRAEEXW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/TISETECVQ8ZY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/CICQLRNVABRT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/AOJRFFGDXYLH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
CT.S3.PR.2 | CIS AWS Benchmark 1.4 3.6  
NIST 800-53 Rev 5 AC-2(4)  
NIST 800-53 Rev 5 AC-4(26)  
NIST 800-53 Rev 5 AC-6(9) | • Establish logging and monitoring | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/QUYPCYYEALDZ  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/FHMUCMMRQQQS  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/SCXNIMSHDDFG

| Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
south-1::control/IXZSFQFPXBPB | • Middle East (Bahrain)  
ar:n:aws:controltower:me-south-1::control/CCLRZDGTBJQX | | |
• Israel (Tel Aviv)  
ar:n:aws:controltower:il-central-1::control/CXDTNMKWUDQA | | |
• Europe (Zurich)  
ar:n:aws:controltower:eucentral-2::control/WNGIRGIYRUQC | | |
• Europe (Spain)  
ar:n:aws:controltower:eusouth-2::control/AGJXNIDFTLTH | | |
• Asia Pacific (Hyderabad)  
ar:n:aws:controltower:ap-south-2::control/JAQQLPHVMOTH | | |
• Middle East (UAE)  
ar:n:aws:controltower:me-central-1::control/YJTKDVALYOKX | | |
• Asia Pacific (Melbourne)  
ar:n:aws:controltower:ap-southeast-4::control/ERNMVMJMMUCD | | |

CT.S3.PR.2
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td></td>
<td>west-2::control/LWFWE3SLVXKVY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1:control/FGDZUXUFIPPHF</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2:control/MSRDFAVXQVX</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1:control/BPVHEZBCMBZH</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1:control/LMPPLBVGMUIC</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
<td></td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1:control/NCKAXMYIELC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2:control/DLJVOQJWVPVWK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td></td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1:control/QIGDYVQVHTOY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1:control/FQYYRCCXLSVY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2:control/UTCYUQMQXRC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1:control/EEJURBQMFKYX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>Europe (Paris)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3:control/EHHLHRKQXKEDX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>South America (São Paulo)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>east-1::control/</td>
<td></td>
<td></td>
<td>QODILHSQEUUDJ</td>
</tr>
<tr>
<td>QODILHSQEUUDJ</td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/</td>
<td></td>
<td></td>
<td>WJNFRTPDVEU</td>
</tr>
<tr>
<td>WJNFRTPDVEU</td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong)</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-east-1::control/</td>
<td></td>
<td></td>
<td>OXUZJOSNXUGV</td>
</tr>
<tr>
<td>OXUZJOSNXUGV</td>
<td></td>
<td></td>
<td>Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
<td></td>
<td></td>
<td>USOLSRKTDAXD</td>
</tr>
<tr>
<td>USOLSRKTDAXD</td>
<td></td>
<td></td>
<td>Asia Pacific (Osaka)</td>
</tr>
<tr>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
<td></td>
<td></td>
<td>PAAAXMDTZLCSW</td>
</tr>
<tr>
<td>PAAAXMDTZLCSW</td>
<td></td>
<td></td>
<td>Europe (Milan)</td>
</tr>
<tr>
<td>arn:aws:controltower:eusouth-1::control/</td>
<td></td>
<td></td>
<td>RPRQAYKWHandr</td>
</tr>
<tr>
<td>RPRQAYKWHandr</td>
<td></td>
<td></td>
<td>Africa (Cape Town)</td>
</tr>
<tr>
<td>arn:aws:controltower:af-south-1::control/</td>
<td></td>
<td></td>
<td>HIPGBZKDQSKN</td>
</tr>
<tr>
<td>HIPGBZKDQSKN</td>
<td></td>
<td></td>
<td>Middle East (Bahrain)</td>
</tr>
<tr>
<td>arn:aws:controltower:mesouth-1::control/</td>
<td></td>
<td></td>
<td>FIUQQJAZYYWBP</td>
</tr>
<tr>
<td>FIUQQJAZYYWBP</td>
<td></td>
<td></td>
<td>Israel (Tel Aviv)</td>
</tr>
<tr>
<td>arn:aws:controltower:il-central-1::control/</td>
<td></td>
<td></td>
<td>XNQIMLEGLW0</td>
</tr>
<tr>
<td>XNQIMLEGLW0</td>
<td></td>
<td></td>
<td>Europe (Zurich)</td>
</tr>
<tr>
<td>arn:aws:controltower:eucentral-2::control/</td>
<td></td>
<td></td>
<td>UEZMQHOAKJKC</td>
</tr>
<tr>
<td>UEZMQHOAKJKC</td>
<td></td>
<td></td>
<td>Europe (Spain)</td>
</tr>
<tr>
<td>arn:aws:controltower:eusouth-2::control/</td>
<td></td>
<td></td>
<td>ZDEKKVEJKNZR</td>
</tr>
<tr>
<td>ZDEKKVEJKNZR</td>
<td></td>
<td></td>
<td>Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td>arn:aws:controltower:apsouth-2::control/</td>
<td></td>
<td></td>
<td>DJANZMGQCQEVC</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CT.S3.PR.3</td>
<td></td>
<td></td>
<td>Middle East (UAE) arn:aws:controltower:me-central-1::control/TQKVAEGMISUA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/HUJNXXTHLDBM</td>
</tr>
<tr>
<td>CT.S3.PR.3</td>
<td>NIST 800-53 Rev 5 AU-9(2)</td>
<td>Optimize costs</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/WNDCHAPIOWPX</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-10</td>
<td>Improve availability</td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/NMQZFXOJBMEI</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/GGFPMXFOJMKL</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-9</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/TWWWTNVBNUNJI</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/RRBBSRTVOULJ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/MGLDCUIJSXOP</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 10.5.3</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ BGAFYFTRFRWB</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 10.5.4</td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ PPCZRHIWDCO</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ UVIJGHTLYXYJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ VKFFQEXOWYRO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ OGNPSHWCNUXG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ ZXXKHALOBBMC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ ZZWZQEUOXAHO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ KEOGAUVAGIWC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ AKENCNOJFZIY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ PMCZHCYFLDME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ JWYSSJGWQIIQI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ TRWVVPDNBLSG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ GDPYJXCBMBR5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Europe (Milan)  | arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.S3.PR.4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CT.S3.PR.4**

**Control identifier**
- CT.S3.PR.4

**Framework**
- NIST 800-53 Rev 5 CA-7
- NIST 800-53 Rev 5 SI-3(8)

**Control objective**
- Establish logging and monitoring

**Control API identifiers, by Region**
- US East (N. Virginia)
  arn:aws:controltower:us-east-1::control/
  YBVXNKBMJYZ0
- US East (Ohio)
  arn:aws:controltower:us-east-1::control/
  YBVXNKBMJYZ0
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td>SI-4</td>
<td></td>
<td>east-2::control/ NBXPLAFUPVPO</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td>SI-4(4)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ OQTGURFVOIFX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>11.5</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ ZEZHVRLECUPJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ TJVIPRFLQFJNU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ CQYPJBFGRVNR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ GVTLGRKRYXAE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ HMPSWNZQRQVD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ ODQFNKZVQAVERW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ JRZYMZVUABUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ HDFMAXJXVUNX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ NGFEGYCAZMOQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ JIKPKVBCQTWZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>west-3::control/</td>
<td></td>
<td></td>
<td>FWDLBIIQYIXW</td>
</tr>
<tr>
<td>• South America</td>
<td></td>
<td></td>
<td>(São Paulo) arn:aws:controltower:sa-east-1::control/BLCIONDMMXME</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td>California) arn:aws:controltower:us-west-1::control/AKPXSEGLBYY</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Hong Kong) arn:aws:controltower:ap-east-1::control/RKMCRSKNQDMF</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Jakarta) arn:aws:controltower:ap-southeast-3::control/XWCENVGPFKV</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>(Osaka) arn:aws:controltower:ap-northeast-3::control/QPOADHZNFITP</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/RQQQDSQMSXSE</td>
</tr>
<tr>
<td>• Africa (Cape</td>
<td></td>
<td></td>
<td>Town) arn:aws:controltower:af-south-1::control/ATMHOATOZICQ</td>
</tr>
<tr>
<td>• Middle East</td>
<td></td>
<td></td>
<td>(Bahrain) arn:aws:controltower:me-south-1::control/HZZGGKDJSMKW</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/NKSQUEXXOSX</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/TGVKWKNMHKGV</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/OWRWJJMCRCQQ</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>CT.S3.PR.5</td>
<td></td>
<td></td>
<td>• Enforce least privilege</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td></td>
<td>• US East (N. Virginia) arn:aws:controlltower:us-east-1::control/ IICSISZKWBP</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>• US East (Ohio) arn:aws:controlltower:us-east-2::control/ QOSGJSIWYQBY</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controlltower:us-west-2::control/ OGTJUXQ8TOQ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>• Canada (Central) arn:aws:controlltower:ca-central-1::control/ RZFOSHDJDWQR</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controlltower:ap-southeast-2::control/ XBIANCEN2NPT</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controlltower:ap-southeast-1::control/ IKFLVYCSQUU</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 7.2.3</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controlltower:eucentral-1::control/ QDUMVXOVRJU</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/ ZQNUYRNLELMH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ KLPNFIYSSKEX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ QTKVX3RHKVWN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ KYICILZPASAZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ OUJUGYPGPBTR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ SKXEZLRKRIF0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ UHGKAZQPWHYD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ EIXEG5OMQVMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ LQTJJUFUBRCL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ ONLMVXXNYTOO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ NKZLNBBHHNYWR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Asia Pacific (Osaka) | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.S3.PR.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT.S3.PR.6</td>
<td></td>
<td>• Optimize costs</td>
<td>• US East (N. Virginia)</td>
</tr>
</tbody>
</table>
|                    |           |                  | arn:aws:controltower:us-
<p>|                    |           |                  | 2114 |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 CP-10</td>
<td></td>
<td>Improve availability</td>
<td>east-1::control/EXSYSTLXGVRE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US East (Ohio)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/SPSBPNJTBBZK</td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/WIBGVDRNDWQR</td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/JVKPEBOPGEKW</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/WVHLLZEVRMQ</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/SIFXKKRHFTFJ</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/HYKSAATXAEUX</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/PTGEDEGHRPD</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/FABQQCRPITA</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/ZDOZUMODNNSH</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/IJYFTUFBPBGPV</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/RPCLZUYKICEAM</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo) |  |  | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northeat-1::control/URWGYOBMHNOV</td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/SUHIETMCJFVC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan) arn:aws:controltower:eu-south-1::control/CTTWTIWREUJJ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/XBKZPYOKFUZT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain) arn:aws:controltower:me-south-1::control/WMOZGCWHFQPY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/IDZZG0PLRKFC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich) arn:aws:controltower:eu-central-2::control/GKFFPPVNGUOY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/IHGNYILKIJRJ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/PCNZKGERWRYG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/WPJQRRMWAVW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/MENFCLCEHUMV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/UIWWHNMZAHUB</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## CT.S3.PR.7

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.S3.PR.7         | • CIS AWS Benchmark 1.4 2.1.1  
• NIST 800-53 Rev 5 AU-9  
• NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-3(6)  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-28  
• NIST 800-53 Rev 5 SC-28(1)  
• NIST 800-53 Rev 5 SC-7(10)  
• NIST 800-53 Rev 5 SI-7(6)  
• PCI DSS version 3.2.1 10.5  
• PCI DSS version 3.2.1 10.5.2  
• PCI DSS version 3.2.1 2.2 | • Encrypt data at rest | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/PCBBDCESRSRLI  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/YVXKKTALMRME  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/IZUDEHGPYPFFF  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/0BNREETGVUFPP  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/NAAPKQTXWJCW  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/LDALEXHCWPWA |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 3.4 8.2.1</td>
<td>• Europe (Frankfurt)</td>
<td><strong>arn:aws:controltower:eu-central-1::control/DRIZXDBSPIDH</strong></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td>• Europe (Ireland)</td>
<td><strong>arn:aws:controltower:eu-west-1::control/DQEFWKNYKCBEB</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (London)</td>
<td><strong>arn:aws:controltower:eu-west-2::control/TIZNPKIQUHWS</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Stockholm)</td>
<td><strong>arn:aws:controltower:eu-north-1::control/NQJSDNASEYTD</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
<td><strong>arn:aws:controltower:ap-south-1::control/XKUUWMRTJL</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Seoul)</td>
<td><strong>arn:aws:controltower:ap-northeast-2::control/SWUYRXFLGS</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
<td><strong>arn:aws:controltower:ap-northeast-1::control/DUXIFIPEQQL</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Paris)</td>
<td><strong>arn:aws:controltower:eu-west-3::control/HTYQAZAMSJ</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo)</td>
<td><strong>arn:aws:controltower:sa-east-1::control/UNCHRCXFVGM</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• US West (N. California)</td>
<td><strong>arn:aws:controltower:us-west-1::control/MFOPWQRPTBK</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Hong Kong)</td>
<td><strong>arn:aws:controltower:ap-east-1::control/OKOMZSSWQSD</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
<td><strong>arn:aws:controltower:ap-</strong></td>
<td></td>
</tr>
</tbody>
</table>
### CT.S3.PR.8

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.S3.PR.8          | • CIS AWS Benchmark 1.4 2.1.2  
• NIST 800-53 Rev 5 AC-17(2)  
• NIST 800-53 Rev 5 AC-4  
• NIST 800-53 Rev 5 IA-5(1)  
• NIST 800-53 Rev 5 SC-12(3)  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-23  
• NIST 800-53 Rev 5 SC-23(3)  
• NIST 800-53 Rev 5 SC-7(4)  
• NIST 800-53 Rev 5 SC-8  
• NIST 800-53 Rev 5 SC-8(1) | • Encrypt data in transit | • US East (N. Virginia)  
ar:aws:controltower:us-east-1::control/  
KQIDXVAURPNL  
• US East (Ohio)  
ar:aws:controltower:us-east-2::control/  
XACDBPRGTYLA  
• US West (Oregon)  
ar:aws:controltower:us-west-2::control/  
MPNBMXJVAAWW  
• Canada (Central)  
ar:aws:controltower:ca-central-1::control/  
GIQAEXNYBIQ  
• Asia Pacific (Sydney)  
ar:aws:controltower:ap-southeast-2::control/  
BYXXCKKRYVIV  
• Asia Pacific (Singapore)  
ar:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td></td>
<td>southeast-1::control/EVSHZCEOCLDQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/RRMXKKLSSMCO</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/WWINGKFVKVZX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/VMEGHEZKVRHV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/VXOWMBPGJWwG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ZWZWDEMHXOH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/KTRFCYEXSTXR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/QWXAZRHGAPOF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/JSFWYVSJEBLH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/DMPZVJVZMNZN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/HXTNWGMJZGM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/QFTBTLBFFSQ</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>southeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KZHGCWHLWF6G0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>northeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RVFBVPTCPJHI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BCLBADAGFADV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Africa (Cape Town)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>STVAP6CPQSUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (Bahrain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BDAB0RJSRZI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Israel (Tel Aviv)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TAZUWKEQXPV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>central-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FXNNHXMTPOB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Spain)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>south-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EGTVZTAIMXV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hyderabad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>south-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>XQMJUIQTGYX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Middle East (UAE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ASXITBMOMFBS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Melbourne)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>southeast-4::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>QPPBAOJDQLF4G</td>
</tr>
</tbody>
</table>
## CT.S3.PR.9

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.S3.PR.9         | NIST 800-53 Rev 5 CP-6(2) | Protect data integrity | **US East (N. Virginia)**
|                    |           |                   | arn:aws:controltower:us-east-1::control/YCXCE2ZFPPPEZ |
|                    |           |                   | **US East (Ohio)**
|                    |           |                   | arn:aws:controltower:us-east-2::control/EQNXZ0TUEHEL |
|                    |           |                   | **US West (Oregon)**
|                    |           |                   | arn:aws:controltower:us-west-2::control/CHIOVWN3WNBWJ |
|                    |           |                   | **Canada (Central)**
|                    |           |                   | arn:aws:controltower:ca-central-1::control/DKOVKUNLSMLJ |
|                    |           |                   | **Asia Pacific (Sydney)**
|                    |           |                   | arn:aws:controltower:ap-southeast-2::control/MOCNFRG1VQIQ |
|                    |           |                   | **Asia Pacific (Singapore)**
|                    |           |                   | arn:aws:controltower:ap-southeast-1::control/HHIBSFOWA0FQ |
|                    |           |                   | **Europe (Frankfurt)**
|                    |           |                   | arn:aws:controltower:eu-central-1::control/DOSCMG5E3QB |
|                    |           |                   | **Europe (Ireland)**
|                    |           |                   | arn:aws:controltower:eu-west-1::control/XXUTOYAX3ZIOM |
|                    |           |                   | **Europe (London)**
|                    |           |                   | arn:aws:controltower:eu-west-2::control/KTUXUGJXYVE0 |
|                    |           |                   | **Europe (Stockholm)**
|                    |           |                   | arn:aws:controltower:eu-north-1::control/NMEVAZFEZWLWD |
|                    |           |                   | **Asia Pacific (Mumbai)**
<p>|                    |           |                   | arn:aws:controltower:ap-south-1::control/YDWXM6GWK3DO |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/GEANURJHOSJU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/HSMBYWJWJVPN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/EQCSOZEMFEOS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/HXMAUGJKIZZI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/MMPITMGATTGG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/KQCIATXNKEVO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/KMRGDQJCKIKH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/DARIPVIIRWUC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/QUSQFJKIGLXK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/MCGZBCKMXTXN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/XDMGVMGSZSXJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>CT.SAGEMAKER.PR.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-21</td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ NXBXUFEBMYSV</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ BILKEGTPYTLO</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ BAYFKDFICPXR</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ ZXUCDQEYSHPT</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CT.SAGEMAKER.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Winter Central-1::control/ AKLAOZHIFAHX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Zurich) arn:aws:controltower:eu-central-2::control/ CBYRCKJUEIW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Spain) arn:aws:controltower:eu-south-2::control/ HPMQGMDDKGSQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/ WVTLJTCJUFXM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Middle East (UAE) arn:aws:controltower:me-central-1::control/ QWHWEDAKQJE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/ MNQSVFBXKDQ</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td>southeast-2::control/KJAHYCKSSRAA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controlltower:ap-southeast-1::control/HLYGAUWTYVVRB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controlltower:eu-central-1::control/FSUJOVACPHJM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controlltower:eu-west-1::control/XRTOTOYPMYBC</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controlltower:eu-west-2::control/NFMRNEASBICM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controlltower:eu-north-1::control/EECQVSZWLOXW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controlltower:ap-south-1::control/EOQDTUMRBXGY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controlltower:ap-northeast-2::control/IHHCXMIJDPUY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controlltower:ap-northeast-1::control/KABHYKBSJCDD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controlltower:eu-west-3::control/KVUSHDIZGCFC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controlltower:saeast-1::control/PKJBXQYFTERYT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controlltower:us-west-1::control/XKYIJGULCHJI</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ UOYFCMRGSGQX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ POCEBHEMGHXJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ GLVFJP0FPXNJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/ XBMHGBXUTPTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/ QMVMDVZDKWA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/ SQBNRAQIOBDV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/ GUOCCAAMNEK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>arn:aws:controltower:eu-central-2::control/ XZRBQYWMHOOL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>arn:aws:controltower:eu-south-2::control/ FFOCTNBPTMAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/ WXNSWEWICMMI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/ QVLCLNWVCVQZ</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Asia Pacific (Melbourne) | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.SAGEMAKER.PR.2  | • NIST 800-53 Rev 5 AC-21  
|                    | • NIST 800-53 Rev 5 AC-3   
|                    | • NIST 800-53 Rev 5 AC-3(7) 
|                    | • NIST 800-53 Rev 5 AC-4   
|                    | • NIST 800-53 Rev 5 AC-4(21) 
|                    | • NIST 800-53 Rev 5 AC-6   
|                    | • NIST 800-53 Rev 5 SC-7   
|                    | • NIST 800-53 Rev 5 SC-7(11) 
|                    | • NIST 800-53 Rev 5 SC-7(16) 
|                    | • NIST 800-53 Rev 5 SC-7(20) 
|                    | • NIST 800-53 Rev 5 SC-7(21) 
|                    | • NIST 800-53 Rev 5 SC-7(3)  
|                    | • NIST 800-53 Rev 5 SC-7(4)  
|                    | • NIST 800-53 Rev 5 SC-7(9)  
|                    | • PCI DSS version 3.2.1 1.2.1 
|                    | • PCI DSS version 3.2.1 1.3  
|                    | • PCI DSS version 3.2.1 1.3.1  
|                    | • PCI DSS version 3.2.1 1.3.2  
|                    | • PCI DSS version 3.2.1 1.3.4  
|                    | • Limit network access  
|                    | • US East (N. Virginia)  
|                    | • US East (Ohio)  
|                    | • US West (Oregon)  
|                    | • Canada (Central)  
|                    | • Asia Pacific (Sydney)  
|                    | • Asia Pacific (Singapore)  
|                    | • Europe (Frankfurt)  
|                    | • Europe (Ireland)  
|                    | • Europe (London)  
|                    | • Europe (Stockholm)  
|                    | arn:aws:controltower:us-east-1::control/BFRWUSCUMZGR 
|                    | arn:aws:controltower:us-east-2::control/GJVQNCVKMEEK 
|                    | arn:aws:controltower:us-west-2::control/OXZGZLYZZDX 
|                    | arn:aws:controltower:ca-central-1::control/MCLENUHMFXO 
|                    | arn:aws:controltower:ap-southeast-2::control/WHZCFXEMTXVF 
|                    | arn:aws:controltower:ap-southeast-1::control/FTKAOANGNFD 
|                    | arn:aws:controltower:eu-central-1::control/SSA0HZ0HNKPC 
|                    | arn:aws:controltower:eu-west-1::control/YEXOMFOQIRQX 
|                    | arn:aws:controltower:eu-west-2::control/BCODMJVNRAYE 
|                    | arn:aws:controltower:eu-north-1::control/SCTOGVXJHSTW 
<p>| southeeast-4::control/QNAPKIIXFEPP |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ZTZRMHDPVQSR</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/AJUBXQWZCNZA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/XRQNSQZACILS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/TETZPAGYDAJQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/YDBJTBNNIMH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/XGNLGNXWPWUFA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/NYSYXCRATNBX</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/PYUYQRAJTEYR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/WLJGIIWFSDQDK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/FQOCOBHBFWNB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/ACIIIOUHBVEN</td>
<td></td>
</tr>
</tbody>
</table>
| | Middle East (Bahrain) | arn:aws:controltower:me-
### Control identifier | Framework | Control objective | Control API identifiers, by Region
---|---|---|---
CT.SAGEMAKER.PR.3 | • NIST 800-53 Rev 5 AC-2(1)  
• NIST 800-53 Rev 5 AC-3(15)  
• NIST 800-53 Rev 5 AC-3(7)  
• NIST 800-53 Rev 5 AC-6  
• NIST 800-53 Rev 5 AC-6(10)  
• NIST 800-53 Rev 5 AC-6(2)  
• PCI DSS version 3.2.1 2.2 | • Enforce least privilege | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/HQQUZBNWPSK0  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/MDQQIVONAGJQ  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/PQOZPZGZVPTNP  
• Canada (Central)  
ar

- Israel (Tel Aviv)
  arn:aws:controltower:il-central-1::control/RBCEKQMEPPSR
- Europe (Zurich)
  arn:aws:controltower:eu-central-2::control/ORWQSODDZ2FL
- Europe (Spain)
  arn:aws:controltower:eu-south-2::control/DFQMNQMGNELE
- Asia Pacific (Hyderabad)
  arn:aws:controltower:ap-south-2::control/FWLXIOSGXLZD
- Middle East (UAE)
  arn:aws:controltower:me-central-1::control/BPGXJEPKEPMPW
- Asia Pacific (Melbourne)
  arn:aws:controltower:ap-southeast-4::control/PZIBDNEVGZNV
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td>central-1::control/ PFWXJZLGHZHS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.2</td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ WFSJGEHTGUJD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.1</td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ BORBTTFXSVMR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ XGSJWJUCDSDD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.1.1</td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ QUAMXLBEEBM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>•</td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ FFLFBLHYHRKH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>•</td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ VECRJW5QQKJX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>•</td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ MWSRFJCKCMYZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>•</td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ YKZDB0CKKZBP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>•</td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ HCJSGYPXGJZD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>•</td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/ NRPXZDRCFVEM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>•</td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/ YHWVCTSC0ZPQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>•</td>
<td>US West (N. California) arn:aws:controltower:us-west-2::control/ MVXZD0CKKZBP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>----------------------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>arn:aws:controltower:us-west-1::control/ JLDGSDBFPEIG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ MAMFMPGXSSSQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/ JGRCEFMTNSLL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/ ODRNGAVPVJLJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/ YIEFICYWYZLU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/ HRMNWAHHHTFT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/ KNZFXFKXDIKD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/ JMEFHEDEXBAE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td>arn:aws:controltower:eucentral-2::control/ OYPVSQELUEML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td>arn:aws:controltower:eusouth-2::control/ TVCLBFAMYSLY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/ RYVVXMPG1YGK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Middle East (UAE)</td>
<td>arn:aws:controltower:me-central-1::control/ JJNUJERNVJM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2131
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.SQS.PR.1        | None      | • Improve resiliency | • US East (N. Virginia)  
arrn:aws:controltower:us-east-1::control/WHMAMEUKIFGE  
• US East (Ohio)  
arrn:aws:controltower:us-east-2::control/LSZZWCFWWEBV  
• US West (Oregon)  
arrn:aws:controltower:us-west-2::control/LIUFDWXIUULF  
• Canada (Central)  
arrn:aws:controltower:ca-central-1::control/SCSCEOQLMMEP  
• Asia Pacific (Sydney)  
arrn:aws:controltower:ap-southeast-2::control/RBKKAATWLGQVS  
• Asia Pacific (Singapore)  
arrn:aws:controltower:ap-southeast-1::control/OCQCVFYIRMUH  
• Europe (Frankfurt)  
arrn:aws:controltower:eu-central-1::control/OYPHJPNFBWWN  
• Europe (Ireland)  
arrn:aws:controltower:eu-west-1::control/WQDIIQGFBSCM  
• Europe (London)  
arrn:aws:controltower:eu-west-2::control/YGMMIMFHGIHD |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/DALEUNIJKTHY</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/WBEXRIRFKJL</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/HBILQHCZJFAW</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/JHJLYBFELNSZ</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/ITBUKXKLPNH</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/BUBHNTXSNVHH</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/PYPJCABAIACE</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/SBDRRPRISVPPY</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/BVGSH1GLNFAP</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/YWVNSHOTESWX</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/GMPKONEBTSAP</td>
</tr>
</tbody>
</table>
| • Africa (Cape Town) |          |                   | arn:aws:controltower:af-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.SQS.PR.2        |                    | Encrypt data at rest | • US East (N. Virginia)  
arrn:aws:controltower:us-east-1::control/
   JBIAREFXJYH |
|                    |                    |                   | • US East (Ohio)  
arrn:aws:controltower:us-east-2::control/
   RXJEPA05CUYZ |
|                    |                    |                   | • US West (Oregon)  
arrn:aws:controltower:us-
<p>|                    |                    |                   | west-2::control/ |
|                    |                    |                   | DGEFI1BBKNAY |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td></td>
<td>west-2::control/AQUJF3QURUFV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/OQPVUZBBDOKR</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/NWPVUCMYWJLV</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/LLQBEYVB0NPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/DNDLXKJDBGSF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/VIRICUDUZRHZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/MPPVBHPMBSNZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ZIGSJHHCJDFFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/JAGVXXLXSRF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/DTARWAFQOJLJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/XDDGSSVPQAVQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/UWKPBOGUMVAX</td>
</tr>
</tbody>
</table>
| | | | South America (São Paulo) arn:aws:controltower:sa-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>east-1::control/</td>
<td>east-1::control/ DCLDBWPA DGYE</td>
<td>east-1::control/ DCLDBWPA DGYE</td>
<td></td>
</tr>
<tr>
<td>US West (N.</td>
<td>US West (N. California)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>California)</td>
<td>arn:aws:controltower:us-west-1::control/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VLTWPSHSLAKQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HGMWZIVVMLXP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ERHG11DRLPC5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HYGNOVUHPJWO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>Europe (Milan)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-1::control/</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MLKLUIARVDHNN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>Africa (Cape Town)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:af-south-1::control/</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AWUNOCUECLQA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East</td>
<td>Middle East (Bahrain)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUAWNTLATCEP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>Israel (Tel Aviv)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:il-central-1::control/</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PZGOUGUCUHIE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td>Europe (Zurich)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eucentral-2::control/</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RCDCYANDWUAI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td>Europe (Spain)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arn:aws:controltower:eu-south-2::control/</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BAJGOTFZUYNO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>Asia Pacific (Hyderabad)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Hyderabad)</td>
<td>arn:aws:controltower:ap-south-2::control/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LQQXTJHPPSCL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Control API identifiers, by Region

- **Middle East (UAE)**
  - arn:aws:controltower:me-central-1::control/LICZJVRYWOS

- **Asia Pacific (Melbourne)**
  - arn:aws:controltower:ap-southeast-4::control/TCOMVFDNX3HQ

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.STEPFUNCTIONS.PR.1 | NIST 800-53 Rev 5 AC-4(26)  
NIST 800-53 Rev 5 AU-10  
NIST 800-53 Rev 5 AU-12  
NIST 800-53 Rev 5 AU-2  
NIST 800-53 Rev 5 AU-3  
NIST 800-53 Rev 5 AU-6(3)  
NIST 800-53 Rev 5 SC-7(9)  
NIST 800-53 Rev 5 SI-7(8)  
PCI DSS version 3.2.1 10.1  
PCI DSS version 3.2.1 10.2.1  
PCI DSS version 3.2.1 10.2.4  
PCI DSS version 3.2.1 10.2.7  
PCI DSS version 3.2.1 10.3.1  
PCI DSS version 3.2.1 10.3.2 | Establish logging and monitoring | US East (N. Virginia)  
US East (Ohio)  
US West (Oregon)  
Canada (Central)  
Asia Pacific (Sydney)  
Asia Pacific (Singapore)  
Europe (Frankfurt)  
Europe (Ireland) | arn:aws:controltower:us-east-1::control/RARWRIGRBHMP  
arwn:aws:controltower:us-east-2::control/CUOQMOAQVWNG  
arwn:aws:controltower:us-west-2::control/OFNAXQITJES  
arwn:aws:controltower:ca-central-1::control/RJCEUDWEBLJ  
arwn:aws:controltower:ap-southeast-2::control/HYFUIUJ0XQNV  
arwn:aws:controltower:ap-southeast-1::control/CWLRwPTPGVQD  
arwn:aws:controltower:eu-central-1::control/NAMQPORQVXQM  
arwn:aws:controltower:eu-west-1::control/SBVYLFYIIXPP |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/KDFTTNMHFBLL</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/CGAFSUBVZBJH</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/FKXONMCUFEDB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/BSARINIRFYGL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ERZIMFMFCIVL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/JGQQRQVFLGDJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/RILHLBPXXUXU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/ASZQNSJURJXR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Hong Kong) arn:aws:controltower:ap-east-1::control/QIWXQALPOWVWN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Jakarta) arn:aws:controltower:ap-southeast-3::control/OUZWAONSGGYA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/ITIIIX0XNEG6V</td>
</tr>
</tbody>
</table>
| | | | • Europe (Milan) arn:aws:controltower:eu-
### CT.STEPFUNCTIONS.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-1::control/BHEGKQAMVASL</td>
</tr>
</tbody>
</table>

- **Control identifier**
- **Framework**
- **Control objective**
- **Control API identifiers, by Region**
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>AU-12</td>
<td></td>
<td>east-2::control/VXVKISEMRURZ</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>AU-2</td>
<td>AU-6(3)</td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>AU-3</td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/MSXYLKDWPWZRNX</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>AU-6(4)</td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>CA-7</td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/BZAERJZTDLTU</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(9)</td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SI-7(8)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/BUACwBSQOPNW</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.2.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/JHBUuWhFLYyg</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.2.7</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/JKUFGuxYKGW</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.3.1</td>
<td></td>
<td></td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.3.2</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/MUJRXYWSIOUS</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/HMTTWFVYBDUQ</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/FLMFYWQVJDZX</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.3.7</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.3.8</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/YHNWNXVKQZP</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.3.9</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.4.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/FLVJBWFDOHDJ</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.4.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.4.3</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/IXMAHHUSDUVR</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.4.4</td>
<td></td>
<td></td>
<td>Europe (Paris)</td>
</tr>
</tbody>
</table>
| PCI DSS version 3.2.1 10.4.5 | |                  | arn:aws:controltower:eu-
### Control identifier, Framework, Control objective, Control API identifiers, by Region

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>west-3:control/GCMXGSVVIEUT</td>
<td>• South America (São Paulo)</td>
<td></td>
<td>arn:aws:controltower:sa-east-1:control/IBZBYQWSSFBM</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1:control/YANYRDEXXDEH</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1:control/MWCJKKKFLGQI</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3:control/MXXULKHAVUNN</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3:control/TVTNAPHRDRZU</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1:control/ULPOIYIDRFWE</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1:control/TYPGEIBRQHQW</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-south-1:control/LRDZXXUZTAVN</td>
</tr>
<tr>
<td>• Israel (Tel Aviv)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:il-central-1:control/EEEKQLYFIYRRG</td>
</tr>
<tr>
<td>• Europe (Zurich)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2:control/RMLLZIIIHUNYX</td>
</tr>
<tr>
<td>• Europe (Spain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2:control/WTJHRSSHPAKK</td>
</tr>
</tbody>
</table>
## CT.WAF-REGIONAL.PR.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.WAF-REGIONAL.PR.1</td>
<td></td>
<td>Limit network access</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/DWPWBOCJUTTO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/UYFKJE0QVWH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/TEUMMEADETS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/TLTACWUKOMMH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/JGWGMJBDHUF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PCI DSS version 3.2.1 6.6</td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/MBRMIIDGSGSBB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eucentral-1::control/XHV5SBHIROY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:eu-west-1::control/KTKPKIVBNQDE</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:eu-west-2::control/CAGANIVPBXCM</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:eu-north-1::control/AYNFYQXOVCGW</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-south-1::control/PWYDDDJMHVKA</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-northeast-2::control/GMIRRPOYKDSI</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-northeast-1::control/GBLQKIZJUWLI</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:eu-west-3::control/PLLMIJKTGYDC</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:sa-east-1::control/FWZWRHWINCPE</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:us-west-1::control/BODSSYLZFBOR</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-east-1::control/VVYKYKULBINF</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>Arn:aws:controltower:ap-southeast-3::control/ADCMRCCWZPA</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Osaka) |           |                  | Arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT.WAF-REGIONAL.PR.2</td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-2144</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| • NIST 800-53 Rev 5 CM-2 | US East (Ohio) | east-1::control/ | US East (Ohio) arn:aws:controltower:us-east-2::control/
| • PCI DSS version 3.2.1 6.6 | US West (Oregon) | BOCYBRNEMMGS | US West (Oregon) arn:aws:controltower:us-west-2::control/
| | Canada (Central) | EQMCVOTIUSBY | Canada (Central) arn:aws:controltower:ca-central-1::control/
| | Asia Pacific (Sydney) | SYPLGHEQYJ | Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/
| | Asia Pacific (Singapore) | QOMUGMEGSIHK | Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/
| | Europe (Frankfurt) | RMIXORRPHQOD | Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/
| | Europe (Ireland) | DLVXAPJNGVIP | Europe (Ireland) arn:aws:controltower:eu-west-1::control/
| | Europe (London) | ECJHXMXRZBVQ | Europe (London) arn:aws:controltower:eu-west-2::control/
| | Europe (Stockholm) | IRXUIROUYYS | Europe (Stockholm) arn:aws:controltower:eu-north-1::control/
| | Asia Pacific (Mumbai) | TKFTXHEZWGT | Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/
| | Asia Pacific (Seoul) | KOGUYYCCSEJB | Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/
| | Asia Pacific (Tokyo) | QOMUGMEGSIHK | Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>northea-1::control/</td>
<td></td>
<td>northea-1::control/</td>
<td>SNJTGWLCRZLW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• South America</td>
<td>TRZOIIDYBNNA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• US West (N.</td>
<td>XEDCRXBBIYWT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>California)</td>
<td>arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific</td>
<td>YYYCBSISOPTC1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific</td>
<td>OCLEQHMCFSEV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific</td>
<td>HHXUYSUFAPFI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Milan)</td>
<td>QYJWCKLEEOXNL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Africa (Cape</td>
<td>arn:aws:controltower:af-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Town)</td>
<td>PLBWFTQGUYP0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Middle East</td>
<td>DMWNVMCVVLCU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Israel (Tel</td>
<td>MQXFOURKFK1Z</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Aviv)</td>
<td>arn:aws:controltower:il-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Zurich)</td>
<td>OJMMCNHJJKKIC</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
</tbody>
</table>
| CT.WAF.PR.1       |           | Limit network access | • US East (N. Virginia)  
ar.n:aws:controltower:us-east-1::control/  
UNASESNVHVZYZ  
• US East (Ohio)  
ar.n:aws:controltower:us-east-2::control/  
VDXLZKVLPFSH  
• US West (Oregon)  
ar.n:aws:controltower:us-west-2::control/  
RVKEMXJTMXJ  
• Canada (Central)  
ar.n:aws:controltower:ca-central-1::control/  
SEEFJQCYQYKK  
• Asia Pacific (Sydney)  
ar.n:aws:controltower:ap-southeast-2::control/  
LBEQHSDADD  
• Asia Pacific (Singapore)  
ar.n:aws:controltower:ap-southeast-1::control/  
AKUBJ0AGFDTU |

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.WAF.PR.1       | NIST 800-53 Rev 5 CA-9(1)  
NIST 800-53 Rev 5 CM-2  
PCI DSS version 3.2.1  
6.6 | Limit network access | • Europe (Spain)  
ar.n:aws:controltower:eu-south-2::control/  
ZEYPELWNMQGB  
• Asia Pacific (Hyderabad)  
ar.n:aws:controltower:ap-south-2::control/  
RWNPUZVHYNHMY  
• Middle East (UAE)  
ar.n:aws:controltower:me-central-1::control/  
NCJSSEWLPACP  
• Asia Pacific (Melbourne)  
ar.n:aws:controltower:ap-southeast-4::control/  
THIHAAMNYOLM |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/ XVRZCGVSPYIL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/ ELFPBLOGRZYE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ LSGSHIHBLV LX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ IRRDWBVEE VZR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ XANETJVAEOSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ PLNVFAREKZX P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ TXHPC T UX MEXT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ JSPZJEWUVU VF A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ ZVSTKBQVF F MQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ SVRRGEYNFUOM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/ NCRRN MOLK HQE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>southeast-3::control/ KCRFFCFEBPEA</td>
<td>• Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/ XKVNI0EEWELE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka) arn:aws:controltower:ap-northeast-3::control/ XKVNI0EEWELE</td>
<td>• Europe (Milan) arn:aws:controltower:europe-central-1::control/ ZDJOMTGOH0JR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Milan) arn:aws:controltower:europe-central-1::control/ ZDJOMTGOH0JR</td>
<td>• Africa (Cape Town) arn:aws:controltower:af-south-1::control/ GQFGYLKZJIDA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town) arn:aws:controltower:af-south-1::control/ GQFGYLKZJIDA</td>
<td>• Middle East (Bahrain) arn:aws:controltower:me-central-1::control/ GPIOUFKAZYRX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain) arn:aws:controltower:me-central-1::control/ GPIOUFKAZYRX</td>
<td>• Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/ BLSNAPKCEBE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/ BLSNAPKCEBE</td>
<td>• Europe (Zurich) arn:aws:controltower:europe-central-2::control/ EUOUIEIKSRQK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich) arn:aws:controltower:europe-central-2::control/ EUOUIEIKSRQK</td>
<td>• Europe (Spain) arn:aws:controltower:europe-south-2::control/ OVMOWJDWSHZX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Spain) arn:aws:controltower:europe-south-2::control/ OVMOWJDWSHZX</td>
<td>• Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/ ZFNFRZNGHWKO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/ ZFNFRZNGHWKO</td>
<td>• Middle East (UAE) arn:aws:controltower:me-central-1::control/ BHBCIBBWQQLS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE) arn:aws:controltower:me-central-1::control/ BHBCIBBWQQLS</td>
<td>• Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/ CMRLTLXFKCK</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## CT.WAF.PR.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| CT.WAF.PR.2        | • NIST 800-53 Rev 5 AC-4(21)  
                   | • NIST 800-53 Rev 5 SC-7  
                   | • NIST 800-53 Rev 5 SC-7(11)  
                   | • NIST 800-53 Rev 5 SC-7(16)  
                   | • NIST 800-53 Rev 5 SC-7(21)  
                   | • PCI DSS version 3.2.1 6.6 | • Limit network access |
|                    | • US East (N. Virginia)  
                   | arn:aws:controltower:us-east-1::control/EVTLDFVJREHX | |
|                    | • US East (Ohio)  
                   | arn:aws:controltower:us-east-2::control/TESBXHDTYCVR | |
|                    | • US West (Oregon)  
                   | arn:aws:aws:controltower:us-west-2::control/WHEYGYOKBFEE | |
|                    | • Canada (Central)  
                   | arn:aws:controltower:ca-central-1::control/WKA0AEFQ8GB | |
|                    | • Asia Pacific (Sydney)  
                   | arn:aws:controltower:ap-southeast-2::control/ZWQCPIXANTTE | |
|                    | • Asia Pacific (Singapore)  
                   | arn:aws:controltower:ap-southeast-1::control/VULAATRXXMNA | |
|                    | • Europe (Frankfurt)  
                   | arn:aws:controltower:eu-central-1::control/QVEBBPUQHHTP | |
|                    | • Europe (Ireland)  
                   | arn:aws:controltower:eu-west-1::control/LAPESUGIQGRW | |
|                    | • Europe (London)  
                   | arn:aws:controltower:eu-west-2::control/FHTQFTBMJHIC | |
|                    | • Europe (Stockholm)  
                   | arn:aws:controltower:eu-north-1::control/HAFOBOPSWQF | |
|                    | • Asia Pacific (Mumbai)  
<pre><code>               | arn:aws:controltower:ap-south-1::control/TZFBNAETOMJR | |
</code></pre>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/HWDELIEBVFK</td>
<td>Framework control identifier</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/WGLOYOZAYLU</td>
<td>Framework control identifier</td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/XQQXGFIHZHQZ</td>
<td>Framework control identifier</td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/GHWIKZXTBTPW</td>
<td>Framework control identifier</td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/NSQKESMWPRQ</td>
<td>Framework control identifier</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td>arn:aws:controltower:ap-east-1::control/WVKFSXCMOAFB</td>
<td>Framework control identifier</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td>arn:aws:controltower:ap-southeast-3::control/BWSNHISVRCNV</td>
<td>Framework control identifier</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td>arn:aws:controltower:ap-northeast-3::control/TLGQQB0RE0SV</td>
<td>Framework control identifier</td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td>arn:aws:controltower:eu-south-1::control/JNFJCDG8QAQJ</td>
<td>Framework control identifier</td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td>arn:aws:controltower:af-south-1::control/OXORBCQFWTQ</td>
<td>Framework control identifier</td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td>arn:aws:controltower:me-south-1::control/IQGCTI0QKDWL</td>
<td>Framework control identifier</td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td>arn:aws:controltower:il-</td>
<td>Framework control identifier</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.WAFV2.PR.1      | • NIST 800-53 Rev 5 AC-4(21)  
|                    | • NIST 800-53 Rev 5 SC-7  
|                    | • NIST 800-53 Rev 5 SC-7(11)  
|                    | • NIST 800-53 Rev 5 SC-7(16)  
|                    | • NIST 800-53 Rev 5 SC-7(21)  
|                    | • PCI DSS version 3.2.1  
|                    | • Limit network access  | • US East (N. Virginia)  
|                    |                        | arn:aws:controltower:us-east-1::control/ XEONMNFZMEEF  
|                    |                        | • US East (Ohio)  
|                    |                        | arn:aws:controltower:us-east-2::control/ JWKJRMTPOQYGL  
|                    |                        | • US West (Oregon)  
|                    |                        | arn:aws:controltower:us-west-2::control/ KYYOLAPACMCW  
|                    |                        | • Canada (Central)  
|                    |                        | arn:aws:controltower:ca-central-1::control/ NPVWKMPMPGNPQ  
|                    |                        | • Asia Pacific (Sydney)  
|                    |                        | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>southeeast-2::control/ITRZPFHNHOVU</td>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/GXKAGSHEBLQA</td>
</tr>
<tr>
<td></td>
<td>• Europe (Frankfurt)</td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/TJZLGRPBEFZZ</td>
</tr>
<tr>
<td></td>
<td>• Europe (Ireland)</td>
<td></td>
<td>arn:aws:aws:controltower:eu-west-1::control/ACZQQJXMSSKX</td>
</tr>
<tr>
<td></td>
<td>• Europe (London)</td>
<td></td>
<td>arn:aws:aws:controltower:eu-west-2::control/WNZNWBPATAGS</td>
</tr>
<tr>
<td></td>
<td>• Europe (Stockholm)</td>
<td></td>
<td>arn:aws:aws:controltower:eu-north-1::control/PQBMKWIWABGD</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td>arn:aws:aws:controltower:ap-south-1::control/ZJTWMPTTRZRR</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td>arn:aws:aws:controltower:ap-northeast-2::control/FLFTSMKDQPBA</td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td>arn:aws:aws:controltower:ap-northeast-1::control/QPSOONUAQWAOR</td>
</tr>
<tr>
<td></td>
<td>• Europe (Paris)</td>
<td></td>
<td>arn:aws:aws:controltower:eu-west-3::control/RQXLAERBFHA</td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo)</td>
<td></td>
<td>arn:aws:aws:controltower:sa-east-1::control/OBDIKEAZGTAV</td>
</tr>
<tr>
<td></td>
<td>• US West (N. California)</td>
<td></td>
<td>arn:aws:aws:controltower:us-west-1::control/NIGBBACJPCNN</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective, by Region</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------------------</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hong Kong)</td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/ALLKEZAETGKXV</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Jakarta)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/WRSGKTAMFUNN</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Osaka)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/ATYEMLPCRYL</td>
<td></td>
</tr>
<tr>
<td>Europe (Milan)</td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/JUDFDFZUPGDV</td>
<td></td>
</tr>
<tr>
<td>Africa (Cape Town)</td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/AFMIFZSFCJAM</td>
<td></td>
</tr>
<tr>
<td>Middle East (Bahrain)</td>
<td></td>
<td>arn:aws:controltower:me-south-1::control/UGZFHMKFGWJ</td>
<td></td>
</tr>
<tr>
<td>Israel (Tel Aviv)</td>
<td></td>
<td>arn:aws:controltower:il-central-1::control/DWWMENQHGMKJ</td>
<td></td>
</tr>
<tr>
<td>Europe (Zurich)</td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/CRGQEGNYRWJ</td>
<td></td>
</tr>
<tr>
<td>Europe (Spain)</td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/THNACPIXQEGW</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Hyderabad)</td>
<td></td>
<td>arn:aws:controltower:ap-south-2::control/HHWUXJHWSTHW</td>
<td></td>
</tr>
<tr>
<td>Middle East (UAE)</td>
<td></td>
<td>arn:aws:controltower:me-central-1::control/LSQMTYNBGDHH</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Melbourne)</td>
<td></td>
<td>arn:aws:controltower:ap-2154</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| CT.WAFV2.PR.2      | • NIST 800-53 Rev 5 CA-9(1)  
|                    | • NIST 800-53 Rev 5 CM-2  
|                    | • PCI DSS version 3.2.1 6.6 | • Limit network access | • US East (N. Virginia)  
|                    |                        |                                | arn:aws:controltower:us-east-1::control/BUwSZUC0YKVA  
|                    |                        |                                | • US East (Ohio)  
|                    |                        |                                | arn:aws:controltower:us-east-2::control/JLGZGQJYB  
|                    |                        |                                | • US West (Oregon)  
|                    |                        |                                | arn:aws:controltower:us-west-2::control/TMPY6M2E0I  
|                    |                        |                                | • Canada (Central)  
|                    |                        |                                | arn:aws:controltower:ca-central-1::control/LZIRMZWBLY  
|                    |                        |                                | • Asia Pacific (Sydney)  
|                    |                        |                                | arn:aws:controltower:ap-southeast-2::control/HGYKZRFIFS  
|                    |                        |                                | • Asia Pacific (Singapore)  
|                    |                        |                                | arn:aws:controltower:ap-southeast-1::control/GITZQTTINV  
|                    |                        |                                | • Europe (Frankfurt)  
|                    |                        |                                | arn:aws:controltower:europe-central-1::control/VDAEULAPS  
|                    |                        |                                | • Europe (Ireland)  
|                    |                        |                                | arn:aws:controltower:europe-west-1::control/FBEJWAWUG  
|                    |                        |                                | • Europe (London)  
|                    |                        |                                | arn:aws:controltower:europe-west-2::control/JEGHUQJDAJ  
|                    |                        |                                | • Europe (Stockholm)  
|                    |                        |                                | arn:aws:controltower:europe-north-1::control/CXJMDJZHU  
<p>|</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/FQITWYCBKFGZ</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/YUWWRBUJCKWS</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/UKLHGBFDOUPK</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/KJZYDZBLLWOQ</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/WKXZFSDZWWS</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/CPNWTODASNRJ</td>
</tr>
<tr>
<td>• Asia Pacific (Hong Kong)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/OLRFCEMWLWYF</td>
</tr>
<tr>
<td>• Asia Pacific (Jakarta)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-3::control/UFZYEYHGRGQ</td>
</tr>
<tr>
<td>• Asia Pacific (Osaka)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-3::control/EVFGOLAANCYD</td>
</tr>
<tr>
<td>• Europe (Milan)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/RGBKCDWRDJD0</td>
</tr>
<tr>
<td>• Africa (Cape Town)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:af-south-1::control/FACNTUGQMNFD</td>
</tr>
<tr>
<td>• Middle East (Bahrain)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:me-</td>
</tr>
</tbody>
</table>
### Control identifier | Framework | Control objective | Control API identifiers, by Region
---|---|---|---
SH.ACM.1 | NIST 800-53 Rev 5 SC-28(3) | NIST 800-53 Rev 5 SC-7(16) | PCI DSS version 3.2.1 4.1 | Encrypt data in transit | US East (N. Virginia) | US East (Ohio) | US West (Oregon) | Canada (Central)

-Israel (Tel Aviv) arn:aws:controltower:il-central-1::control/KBVPZPMKGBZM
-Europe (Zurich) arn:aws:controltower:eu-central-2::control/SSUYGEPFPPHT
-Europe (Spain) arn:aws:controltower:eu-south-2::control/WYZTRLPDPGB
-Asia Pacific (Hyderabad) arn:aws:controltower:ap-south-2::control/EBHCUTSNUBLCE
-Middle East (UAE) arn:aws:controltower:me-central-1::control/OCOJDDRHTCD
-Asia Pacific (Melbourne) arn:aws:controltower:ap-southeast-4::control/AUFYBCLSXDVR

### SH.ACM.1

- NIST 800-53 Rev 5 SC-28(3)
- NIST 800-53 Rev 5 SC-7(16)
- PCI DSS version 3.2.1 4.1
- Encrypt data in transit
- US East (N. Virginia) arn:aws:controltower:us-east-1::control/MBCMJMZJGLS
- US East (Ohio) arn:aws:controltower:us-east-2::control/WXFMEFMVLUM
- US West (Oregon) arn:aws:controltower:us-west-2::control/ANXKMBZMHTME
- Canada (Central) arn:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>central-1::control/BUYCUVITYPVG</td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/PKBECM2KORWL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/MSYJENQMXJXK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/VYLDSDKRLKKA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ZKVPMZUWFJ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/BKDDMYWNGSH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/QPVAASCWYZA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/VPNPOEIIUJA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/TFNTRZCUNKTO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/EYLAEGFQCWO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/EJGKLXXBAEET</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/OWLPOEUMXJ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California) arn:aws:controltower:ap-northeast-1::control/WEALVUOVCW</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2158
### SH.APIGateway.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.APIGateway.1    | • NIST 800-53 Rev 5 AC-4(26)  
• NIST 800-53 Rev 5 AU-10  
• NIST 800-53 Rev 5 AU-12  
• NIST 800-53 Rev 5 AU-2  
• NIST 800-53 Rev 5 AU-3  
• NIST 800-53 Rev 5 AU-6(3)  
• NIST 800-53 Rev 5 AU-6(4)  
• NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 SC-7(9)  
• NIST 800-53 Rev 5 SI-7(8)  
• PCI DSS version 3.2.1 10.1  
• PCI DSS version 3.2.1 10.2.1  
• PCI DSS version 3.2.1 10.2.4  
• PCI DSS version 3.2.1 10.2.7  
• PCI DSS version 3.2.1 10.3.1  
• PCI DSS version 3.2.1 10.3.2  
• PCI DSS version 3.2.1 10.3.3  
• PCI DSS version 3.2.1 10.3.4  
• PCI DSS version 3.2.1 10.3.5 | • Establish logging and monitoring | arn:aws:controltower:us-west-1::control/WRLVGHVNTPPDD |
### Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
SH.APIGateway.2 | • PCI DSS version 3.2.1 10.3.6 | • Encrypt data in transit | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/MNPVCMDMYSLZ
| | | | • US East (Ohio) arn:aws:controltower:us-east-2::control/KPORKNTFQEXN
| | | | • US West (Oregon) arn:aws:controltower:us-west-2::control/YPGHQDSXNTVO
| | | | • Canada (Central) arn:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-23(3)</td>
<td></td>
<td>central-1::control/FDWKKFDTZXX</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Asia Pacific (Sydney)
arn:aws:controltower:ap-southeast-2::control/HCPQXEYLPUTG

Asia Pacific (Singapore)
arn:aws:controltower:ap-southeast-1::control/GCOLOAWADDA

Europe (Frankfurt)
arn:aws:controltower:eu-central-1::control/BBRKRQRFKHFMB

Europe (Ireland)
arn:aws:controltower:eu-west-1::control/UADGCPBNJJB5

Europe (London)
arn:aws:controltower:eu-west-2::control/NSPCYWJJBDRH

Europe (Stockholm)
arn:aws:controltower:eu-north-1::control/JJLTDTNZHJXM

Asia Pacific (Mumbai)
arn:aws:controltower:ap-south-1::control/ZWRPSDCJJTPW

Asia Pacific (Seoul)
arn:aws:controltower:ap-northeast-2::control/RDMVJESGQCRW

Asia Pacific (Tokyo)
arn:aws:controltower:ap-northeast-1::control/AZWAXOWQMNJ

Europe (Paris)
arn:aws:controltower:eu-west-3::control/UOSOHIJTFCO

South America (São Paulo)
arn:aws:controltower:saeast-1::control/EJSPZRWREGJ0

US West (N. California)
### SH.APIGateway.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.APIGateway.3    | • NIST 800-53 Rev 5 CA-7  
                     • PCI DSS version 3.2.1 10.1  
                     • PCI DSS version 3.2.1 10.2.1  
                     • PCI DSS version 3.2.1 10.2.4  
                     • PCI DSS version 3.2.1 10.2.7  
                     • PCI DSS version 3.2.1 10.3.1  
                     • PCI DSS version 3.2.1 10.3.2  
                     • PCI DSS version 3.2.1 10.3.3  
                     • PCI DSS version 3.2.1 10.3.4  
                     • PCI DSS version 3.2.1 10.3.5  
                     • PCI DSS version 3.2.1 10.3.6 | • Establish logging and monitoring | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/0BBBFBBHDUVX  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/TSNMTDBRWMPT  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/LQTFJGDAOWKI  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/XCCMVZSAVCXR  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/PMGEOVFNRWUJ  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/TGMMXHEOYZLH  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/BXICHDHLEQE  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/AQQCYZGINFNV  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/OELZOKBLNQGJ  
• Europe (Stockholm)  
arn:aws:controltower:eu-west-3::control/YXQFQWQFVKMK  
<p>|</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.APIGateway.4</td>
<td></td>
<td>Protect configurations, Manage vulnerabilities</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/SJEXUS0XPSPK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/MEGBFCTBLAPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/WGCGXWCPMWBV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>central-1::control/EZACRQZNMTPK</td>
<td>• Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/JDOSDYPYPULQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/ASBMSMGXQKPS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/YYWZOFQOQUU</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/HAVZHQBLTYRE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/RRTYUCALLKNM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ONNVQAVMAVB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/RXPVPYDQHYM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/CWHOFIFMIUUS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/XIGRNZOUFDKF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/HUHCOZDVSPFJ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/TIJDZCZLPKDU</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• US West (N. California)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2164
### SH.APIGateway.5

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.APIGateway.5    | - NIST 800-53 Rev 5 CA-9(1)  
- NIST 800-53 Rev 5 CM-3(6)  
- NIST 800-53 Rev 5 SC-13  
- NIST 800-53 Rev 5 SC-28  
- NIST 800-53 Rev 5 SC-28(1)  
- NIST 800-53 Rev 5 SC-7(10)  
- NIST 800-53 Rev 5 SI-7(6)  
- PCI DSS version 3.2.1 3.4 | - Encrypt data at rest | - US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/IDZLZNCEBCDN  
- US East (Ohio)  
arn:aws:controltower:us-east-2::control/KTVOXDKUCOLS  
- US West (Oregon)  
arn:aws:controltower:us-west-2::control/ZUWFIOFNKCI  
- Canada (Central)  
arn:aws:controltower:ca-central-1::control/UHUYXDUHUCFJ  
- Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/QRFWQMUPBGAL  
- Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/YKQIWSPLFHMH  
- Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/ICJXNXXFNJBJH  
- Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/IHBHKZKMHHEJ  
- Europe (London)  
arn:aws:controltower:eu-west-2::control/CPPQTSZSOZEN  
- Europe (Stockholm)  
arn:aws:controltower:eu-west-3::control/IDZLZNCEBCDN |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.APIGateway.8</td>
<td></td>
<td>Use strong</td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>authentication</td>
<td>arn:aws:controltower:us-east-1::control/UBJNIPLJAUPP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/UQEZDQFKINS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/VOMLOGQZTVTH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>IESMZIOLTCP</td>
<td>central-1::control/</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AOTXAAIXUZGO</td>
</tr>
<tr>
<td>WGRUYTXBKJPT</td>
<td>central-1::control/</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/WGRUYTXBKJPT</td>
</tr>
<tr>
<td>FEMIWYHJXJIR</td>
<td>central-1::control/</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/FEMIWYHJXJIR</td>
</tr>
<tr>
<td>APXJLVKOBSEB</td>
<td>central-1::control/</td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/APXJLVKOBSEB</td>
</tr>
<tr>
<td>FPBVORGPWAIT</td>
<td>central-1::control/</td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/FPBVORGPWAIT</td>
</tr>
<tr>
<td>IUVYTYANIHHU</td>
<td>central-1::control/</td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/IUVYTYANIHHU</td>
</tr>
<tr>
<td>LPEGHVWXGRWR</td>
<td>central-1::control/</td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/LPEGHVWXGRWR</td>
</tr>
<tr>
<td>KXMSHTINJODP</td>
<td>central-1::control/</td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/KXMSHTINJODP</td>
</tr>
<tr>
<td>SDTKOAZRPRVU</td>
<td>central-1::control/</td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/SDTKOAZRPRVU</td>
</tr>
<tr>
<td>EXXWWSPFGNML</td>
<td>central-1::control/</td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/EXXWWSPFGNML</td>
</tr>
<tr>
<td>JOEIBMZJMELB</td>
<td>central-1::control/</td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/JOEIBMZJMELB</td>
</tr>
<tr>
<td>2167</td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:ap-northeast-1::control/2167</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.APIGateway.9    | • NIST 800-53 Rev 5 AC-4(26)  
• NIST 800-53 Rev 5 AU-10  
• NIST 800-53 Rev 5 AU-12  
• NIST 800-53 Rev 5 AU-2  
• NIST 800-53 Rev 5 AU-3  
• NIST 800-53 Rev 5 AU-6(3)  
• NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 SC-7(9)  
• NIST 800-53 Rev 5 SI-7(8)  
• PCI DSS version 3.2.1 10.1  
• PCI DSS version 3.2.1 10.2.1  
• PCI DSS version 3.2.1 10.2.4  
• PCI DSS version 3.2.1 10.2.7  
| • Establish logging and monitoring | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/ RGZSRLZYXNAM  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/ FBDMHIKAKXYQ  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/ ZNUGCJCJBNYR  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/ YSRNXOSPMPAR  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/ XNMGCOKKDRTM  
| • Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/ HSLYQANZTCLF  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/ TCUHJQTYQIXF  
| • Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/ GREKWXLBUMYC  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/ HWHLVPFIFLLT  
• Europe (Stockholm)  
arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td>north-1::control/FSETKHTFGIHY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/TPQDZNHEPZZY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/HUBWBMOQCTCI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/KBYFZPAGEIEY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/IWTDNHA0VXWZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/OCKVZREZIQQW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/EIJKPXWMUFUM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Account.1</td>
<td>• CIS AWS Benchmark 1.4 1.2</td>
<td>• Establish logging and monitoring</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/RYAEGDBVYSEZ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/DDPIKUWOUKYYH</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2(2)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/SCHNXFCTFZJG</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>central-1::control/IBIPKUPTGKCH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/TDCHULZQJJYU</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/CLWAFAIJBWU0</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eucentral-1::control/LOEEHDRVHTWK</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:euseast-1::control/TUJJHFEUQRTD</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:euseast-2::control/PZTQKPPLUJQC</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eunorth-1::control/ACKPBIJEHTKJO</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/IDEZUGBAFKOT</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/EMCJNTXLMTC5</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/DTWMAHWPBVQX</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:euseast-3::control/KIIMXDNABIMQ</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/KQCHSPVLTIWQ</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.Athena.1</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Encrypt data at rest</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/MABMXMDUIFUT</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/CDSTYMRKVGCQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/STTPACNBPVWU</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/AWVBRDFAUVVV</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/CTKQUVOALPDZ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/SMPSMIXYUYOT</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/HHBBYUMI8BF</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/AADNNDJBSLRZ</td>
</tr>
</tbody>
</table>
|                    |                     |                      | • Europe (Stockholm) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.AutoScaling.1    |           | • Improve availability | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/ZWORVQKMSV
|                     |           |                  | • US East (Ohio) arn:aws:controltower:us-east-2::control/JQMVONDOPRIM |
|                     |           |                  | • US West (Oregon) arn:aws:controltower:us-west-2::control/CCTCKXMLWFDI |
|                     |           |                  | • Canada (Central) arn:aws:controltower:ca- |

### SH.AutoScaling.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.AutoScaling.1    |           | • Improve availability | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/ZWORVQKMSV
<p>|                     |           |                  | • US East (Ohio) arn:aws:controltower:us-east-2::control/JQMVONDOPRIM |
|                     |           |                  | • US West (Oregon) arn:aws:controltower:us-west-2::control/CCTCKXMLWFDI |
|                     |           |                  | • Canada (Central) arn:aws:controltower:ca- |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>central-1::control/IKKVSEWGLMB</td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/HPBROJJRPRQP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/XDFGVGCIRJGE</td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/GVPMLVRGBXZI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/BVVBNBHSIGXC</td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/SZIHWNZUDWBU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ZITVBPNZFKCC</td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/RJDQSGGXMFNH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ZAUGAPPKZGCZ</td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/CXXHKFGCSWUH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/LMYDOAQQOYC</td>
<td>• South America (São Paulo) arn:aws:controltower:saeast-1::control/EQAEEUGLWPUQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SH.AutoScaling.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.AutoScaling.2</td>
<td>- NIST 800-53 Rev 5 CP-10</td>
<td>• Improve availability</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/WPQFDZGIKXEN</td>
</tr>
<tr>
<td></td>
<td>- NIST 800-53 Rev 5 CP-2(2)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ZBMNAYFPMEZY</td>
</tr>
<tr>
<td></td>
<td>- NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/TOSTAVLMCDIG</td>
</tr>
<tr>
<td></td>
<td>- NIST 800-53 Rev 5 SC-36</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/JXPAOXBTCBUY</td>
</tr>
<tr>
<td></td>
<td>- NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/CFQPPPGBKHPVE</td>
</tr>
<tr>
<td></td>
<td>- NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/DCTOFQWBSEI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/LOJSJXUWYJIP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/PPNROGQXGMTG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/NJMEPPCINGYY</td>
</tr>
</tbody>
</table>
|                   |                            |                            | • Europe (Stockholm) arn:aws:controltower:eu-
Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
SH.AutoScaling.3 | NIST 800-53 Rev 5 AC-3 | Protect configurations | US East (N. Virginia) arn:aws:controltower:us-east-1::control/HHCYIDJRAZNC
SH.AutoScaling.3 | NIST 800-53 Rev 5 AC-3(15) |  | US East (Ohio) arn:aws:controltower:us-east-2::control/SHYROJUETSEF
SH.AutoScaling.3 | NIST 800-53 Rev 5 AC-3(7) |  | US West (Oregon) arn:aws:controltower:us-west-2::control/SMYBKBKBIJSSJ
SH.AutoScaling.3 | NIST 800-53 Rev 5 AC-6 |  | Canada (Central) arn:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>central-1::control/JWINTSKAZGY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/RSJYIFHSKNSS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/VJQHUCFGCESL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/XZHHUAZPENXE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/CBRREQKVJIEL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/CNFJJCJJGMTZ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/WQMEEXRGQCEX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/LPSGVWSPBFOM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/VMBGAYOTTZMK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/EOCUBAKFMZJT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/OZFKELNIVWUQY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/YEWHZHIZIYNYF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.AutoScaling.4</td>
<td></td>
<td>Protect configurations</td>
<td>arn:aws:controltower:us-west-1::control/EIDKBDTZ0YAF</td>
</tr>
</tbody>
</table>

**SH.AutoScaling.4**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.AutoScaling.4</td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Protect configurations</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/IKDSU0XJFKTR</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/EMOKDNJXDAOB</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2(2)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/AFMJRQPIACSQ</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/CHDIMYUFAZQV</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/HCJSOXALINQW</td>
</tr>
<tr>
<td></td>
<td>7.1.1</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/DJGIZSCSOQYX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/HKOBHUBPPUYQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/RVRAQDXQ0G5R</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/YWCXENXFSYON</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.AutoScaling.6    |           | Improve availability | • US East (N. Virginia)  
ar:aws:controltower:us-east-1::control/  
BTMIFVPUSJBQ  
• US East (Ohio)  
ar:aws:controltower:us-east-2::control/  
GMGXMVALLWEBEA  
• US West (Oregon)  
ar:aws:controltower:us-west-2::control/  
OWWXHTQSYK  
• Canada (Central)  
ar:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>central-1::control/ZFZUXSLEGCXK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/TNSLXHUTKWTZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/HRPNUEA0HJ1V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eucentral-1::control/YVWD5ASKAHEJO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:europe-west-1::control/OMIRAXEILQAK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:europe-west-2::control/TFXWZSADFQWGN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:aws:controltower:europe-north-1::control/GVTHRVBNLZLB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/NACEEPQFGKZT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/WOMTJFIXVURH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/XNJDGLJOHAMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:europe-west-3::control/HXJMEZAVXVYO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:saeast-1::control/LVDJEVJJMV5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.AutoScaling.9</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/SECDBDYCHCAO</td>
</tr>
</tbody>
</table>

### SH.AutoScaling.9

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.AutoScaling.9</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Manage vulnerabilities</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/VEIRXQKRRWAP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/KKACTCVDRBMF</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2(2)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/SKBFDFDSPRRTT</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/DVIOWKWKIFHK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/XFPEFWDNOLGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/EGWOHNMPKMI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/TLYXQULSGYQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/TYNAB0HYGDXR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/RIPUYTWTZYRU</td>
</tr>
</tbody>
</table>
|                    |                        |                      | Europe (Stockholm) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Autoscaling.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **NIST 800-53 Rev 5 AC-21**
- **NIST 800-53 Rev 5 AC-3**
- **NIST 800-53 Rev 5 AC-3(7)**
- **NIST 800-53 Rev 5 AC-4**
- **NIST 800-53 Rev 5 AC-4(21)**
- **NIST 800-53 Rev 5 AC-6**

- **US East (N. Virginia)**
  - ARN: aws:controltower:us-east-1::control/QVGJOLZXDNNY
- **US East (Ohio)**
  - ARN: aws:controltower:us-east-2::control/HLVWEDLVEUES
- **US West (Oregon)**
  - ARN: aws:controltower:us-west-2::control/CQYHNDCSDDOKU
- **Canada (Central)**
  - ARN: aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>central-1::control/IXJDZWNANLKO</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/BRQEJB5GONN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/IXQGONQSFQTY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/VQDIOMARRWE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/NQPKEMKFOZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/NQPKEMKFOZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/DQMIOMPDBGU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/NRPXJATICULB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/IYOEVEYCOV</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/IRMFCHOI0JRAI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/VYV1APRLJUX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:south-america-east-1::control/YPRGQTSFQYWEB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>SH.CloudTrail.1</td>
<td></td>
<td>Establish logging and monitoring</td>
<td>arn:aws:controltower:us-west-1::control/TFRBKGKMRKMW</td>
</tr>
</tbody>
</table>

### SH.CloudTrail.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.CloudTrail.1</td>
<td>CIS AWS Benchmark 1.4 3.1</td>
<td>• Establish logging and monitoring</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/JSPHSKFGKQKQC</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4(26)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/TKEVYJJCXGTL</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6(9)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/RQFYEFHIOOMQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/JOVYOCPUBVQT</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/MEWHPMWRUBQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/PRVEFKIRBUOG</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ZSWMQFPNNOS</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/FBWNWOPLCWCZ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------</td>
<td>--------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>SH.CloudTrail.2</td>
<td>CIS AWS Benchmark 1.4 3.7</td>
<td>Encrypt data at rest</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/TCMJVYRGWMMOM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-9</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/JAYTWZHWWNMB</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/YHMCHLVAJGXW</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/RHDAYMWQSNQM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/GJBSDWACBFQE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/GDBNBTCDAXFU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/PNVPPZDHTLDX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/PLGOVUAZUWPL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/BJLEQOVHKVJH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/BJKTTLYAWQNP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/OESJOHRDGIP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/OMMCJBYCQGMK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/QRDXCEHXZQVS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/FQYLTXGIK00B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SH.CloudTrail.4

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.CloudTrail.4</td>
<td></td>
<td>Protect data integrity</td>
<td>arn:aws:controltower:us-east-1::control/RKYSICJXWUGL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/PQKRKRAZHNNQGQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/RFTWGBDAOZZG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/JVY2ZM2LDDDRJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/RCLUACHBLUEJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/KDJOJWFSTFL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/BAAGHE0YWZSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/OHTOUMWXBHVL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/YMWJQDYNLGYX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-2::control/BDAGHE0YWZSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/BBAGHE0YWZSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-1::control/CAAGHE0YWZSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/CDAAGHE0YWZSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-south-2::control/DAAGHE0YWZSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-4::control/EAAGHE0YWZSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-2::control/FAAGHE0YWZSK</td>
</tr>
</tbody>
</table>

- CIS AWS Benchmark 1.4 3.2
- NIST 800-53 Rev 5 AU-9
- NIST 800-53 Rev 5 SI-4
- NIST 800-53 Rev 5 SI-7(1)
- NIST 800-53 Rev 5 SI-7(3)
- NIST 800-53 Rev 5 SI-7(7)
- PCI DSS version 3.2.1 10.5.5
- PCI DSS version 3.2.1 11.5
- PCI DSS version 3.2.1 2.2

- US East (N. Virginia)
- US East (Ohio)
- US West (Oregon)
- Canada (Central)
- Asia Pacific (Sydney)
- Asia Pacific (Singapore)
- Europe (Frankfurt)
- Europe (Ireland)
- Europe (London)
- Europe (Stockholm)
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.CloudTrail.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Control identifier**: SH.CloudTrail.5

**Framework**

- CIS AWS Benchmark 1.4.3.4
- NIST 800-53 Rev 5 AC-2(4)
- NIST 800-53 Rev 5 AC-4(26)
- NIST 800-53 Rev 5 AC-6(9)
- NIST 800-53 Rev 5 AU-10
- NIST 800-53 Rev 5 AU-12

**Control objective**: Establish logging and monitoring

**Control API identifiers, by Region**

- **US East (N. Virginia)**
  - arn:aws:controltower:us-east-1::control/
    - MSSQQOYZRTEE

- **US East (Ohio)**
  - arn:aws:controltower:us-east-2::control/
    - ZLNHKOHADVEF

- **US West (Oregon)**
  - arn:aws:controltower:us-west-2::control/
    - BGLRJKKRJVM

- **Canada (Central)**
  - arn:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td></td>
<td>central-1::control/ UHRCNDTXYIU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(1)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ WRJBVWCCGCPA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ NZAPCFAOUVS</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-7(1)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ MJUAASUPXJU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-9(7)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ UWGSZEDUWTBP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-20</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ SWWUBFUZIQOR</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ VOKJCUPZNXG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ TNFFXMEKKNU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ RALDKYFCIKBO</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ OBOZWIHZLRL</td>
</tr>
<tr>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/ DZEZVIWHHEMA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/ UTZGEEHNTWL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------</td>
<td>-----------------------------------------------</td>
<td>------------------------------------------------------</td>
</tr>
</tbody>
</table>
| SH.CodeBuild.1     | • NIST 800-53 Rev 5        | • Use strong authentication                  | • US East (N. Virginia) arn:aws:controltower:us-
|                    | 6.4.4                      |                                               | east-1::control/IKRQWXXPR1UG                          |
|                    | • PCI DSS version 3.2.1    |                                               | • US East (Ohio) arn:aws:controltower:us-
|                    | 6.4.4                      |                                               | east-2::control/VTYYJ3LQHCRTP                         |
|                    | • PCI DSS version 3.2.1    |                                               | • US West (Oregon) arn:aws:controltower:us-
|                    | 8.2.1                      |                                               | west-2::control/QVAROLLMZG0A                          |
|                    |                           |                                               | • Canada (Central) arn:aws:controltower:ca-
|                    |                           |                                               | central-1::control/SGUSWAWCFWN                        |
|                    |                           |                                               | • Asia Pacific (Sydney) arn:aws:controltower:ap-
|                    |                           |                                               | southeast-2::control/OGBNWMEGMMN                      |
|                    |                           |                                               | • Asia Pacific (Singapore) arn:aws:controltower:ap-
|                    |                           |                                               | southeast-1::control/ECIANAYMRGIX                     |
|                    |                           |                                               | • Europe (Frankfurt) arn:aws:controltower:eu-
|                    |                           |                                               | central-1::control/GЙYCUMZXBOOX                       |
|                    |                           |                                               | • Europe (Ireland) arn:aws:controltower:eu-
|                    |                           |                                               | west-1::control/KAHXXGGQXKX                           |
|                    |                           |                                               | • Europe (London) arn:aws:controltower:eu-
|                    |                           |                                               | west-2::control/MSIHJHKZRCLN                           |
|                    |                           |                                               | • Europe (Stockholm) arn:aws:controltower:eu-
|                    |                           |                                               | west-2::control/QVAROLLMZG0A                          |
### Control identifier | Framework | Control objective | Control API identifiers, by Region
---|---|---|---

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.CodeBuild.2    |           | Use strong authentication | US East (N. Virginia) arn:aws:controltower:us-east-1::control/SNJJCJYKNUBTW  
|                   |           |                    | US East (Ohio) arn:aws:controltower:us-east-2::control/NEQ1SRQ0QVGV  
|                   |           |                    | US West (Oregon) arn:aws:controltower:us-west-2::control/XKWAJFIWDTOW  
<p>|                   |           |                    | Canada (Central) arn:aws:controltower:ca- |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| central-1::control/YUAEVTSMXVYI | • Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/NENDYNYMGBAL  
• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/UDLMCNZRQMEI  
• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ZRPUUAKXXKYNQ  
• Europe (Ireland) arn:aws:controltower:eu-west-1::control/QSUUNZFUBPXG  
• Europe (London) arn:aws:controltower:eu-west-2::control/AUCSKLF3XSJZ  
• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/VMUOHSUKEGZY  
• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/RBONONXLXNLMM  
• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/IHLKDRJWXUHXX  
• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/AMKMYJENQHDH  
• Europe (Paris) arn:aws:controltower:eu-west-3::control/DZSVNXMIBQOH  
• South America (São Paulo) arn:aws:controltower:saeast-1::control/HNGUINPSTYDX  
• US West (N. California) |
### Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
SH.CodeBuild.3 | NIST 800-53 Rev 5 CA-9(1) | Encrypt data at rest | US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/PSTQZTBWYCPR

|  | NIST 800-53 Rev 5 CM-3(6) |  | US East (Ohio)  
arn:aws:controltower:us-east-2::control/RPWVHXRXWZUU

|  | NIST 800-53 Rev 5 SC-13 |  | US West (Oregon)  
arn:aws:controltower:us-west-2::control/FBFYJYXDGJDO

|  | NIST 800-53 Rev 5 SC-28(1) |  | Canada (Central)  
arn:aws:controltower:ca-central-1::control/NZBZRTLNLRHC

|  | NIST 800-53 Rev 5 SI-7(6) |  | Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/HBOJXAIXDHPV

|  |PCI DSS version 3.2.1 10.5 |  | Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/DMDCAEHSYVYD

|  | PCI DSS version 3.2.1 10.5.2 |  | Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/FCGGEPJWZPS

|  | PCI DSS version 3.2.1 2.2 |  | Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/RROQJCOSEBIO

|  | PCI DSS version 3.2.1 3.4 |  | Europe (London)  
arn:aws:controltower:eu-west-2::control/FUIOFBRISQPW

|  |  |  | Europe (Stockholm)  
arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.CodeBuild.4</td>
<td></td>
<td>Establish logging and monitoring</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/HQVCCVHGQAZB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/HMIMEZDEFGZN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/RUQCXNA0EZOB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td></td>
<td>central-1::control/KWXNGCKXFPFJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td>Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/NQSLJQHQTLKU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td>Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/VMCOHQOCISWJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-9(7)</td>
<td></td>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/FRTPFQMERZYU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/EQRAZACKPKRQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/KFJKYEFMELB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/RAWBSXYDHUPU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4</td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/VZJSHORKBLYN</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/DEBWVFLXAGZI</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/WUOXVMEZAFHA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/LWFGGPAXPFV</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td></td>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/KCHFPNLICCG</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SH.CodeBuild.5

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.CodeBuild.5     | • NIST 800-53 Rev 5 AC-2(1)  
                    • NIST 800-53 Rev 5 AC-3  
                    • NIST 800-53 Rev 5 AC-3(15)  
                    • NIST 800-53 Rev 5 AC-3(7)  
                    • NIST 800-53 Rev 5 AC-5  
                    • NIST 800-53 Rev 5 AC-6  
                    • NIST 800-53 Rev 5 AC-6(10)  
                    • NIST 800-53 Rev 5 AC-6(2)  
                    • PCI DSS version 3.2.1 8.2.1 | • Enforce least privilege |• US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/KMMZLEUZMTDU  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/UNCDCNTDSUE  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/ENIPGUAEXSC  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/HIRXVHWDXNSB  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/CLWTRMGVZRRH  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/EIJPCEZIKHZD  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/UVNIPDHPSIEL  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/GGIGOXGEKTWZ  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/OQLYJKDVWGRQ  
• Europe (Stockholm)  
arn:aws:controltower:eu-west-3::control/ENIPGUAEXSC |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.DMS.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Control identifier**
  - SH.DMS.1

- **Framework**
  - NIST 800-53 Rev 5 AC-21
  - NIST 800-53 Rev 5 AC-3
  - NIST 800-53 Rev 5 AC-3(7)
  - NIST 800-53 Rev 5 AC-4
  - NIST 800-53 Rev 5 AC-4(21)
  - NIST 800-53 Rev 5 AC-6

- **Control objective**
  - Limit network access
  - Enforce least privilege

- **Control API identifiers, by Region**
  - US East (N. Virginia)
    - arn:aws:controltower:us-east-1::control/TJKXNFIKOWSN
  - US East (Ohio)
    - arn:aws:controltower:us-east-2::control/TXOKWLKHFOE
  - US West (Oregon)
    - arn:aws:aws:controltower:us-west-2::control/JPRWVQHPQWQX
  - Canada (Central)
    - arn:aws:controltower:ca-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

central-1::control/AAJEZEBPCQKN

Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/DHOJQSKCRIQH

Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/PZGUNNSCNTBM

Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/UGTQRQUDPFTC

Europe (Ireland) arn:aws:controltower:eu-west-1::control/AIHUAMECLLHS

Europe (London) arn:aws:controltower:eu-west-2::control/UUZLDXFJSJIK

Europe (Stockholm) arn:aws:controltower:eu-north-1::control/XSOOILKWTSYF

Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/GIDDGGAFOUJI

Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/KKTKHLMBETI

Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/MIMZFVUEANJV

Europe (Paris) arn:aws:controltower:eu-west-3::control/FJWDHHBNJGDS

South America (São Paulo) arn:aws:controltower:saeast-1::control/VEFFWEOQXY0SJ

US West (N. California)
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.DynamoDB.1      | • NIST 800-53 Rev 5 CP-10  
                   |           | • Improve availability | • US East (N. Virginia)  
                   |           |                               | arn:aws:controltower:us-east-1::control/  
                   |           |                               | XQGJDGQQKGCRR  
                   |           |                               | • US East (Ohio)  
                   |           |                               | arn:aws:controltower:us-east-2::control/  
                   |           |                               | RIGUOEEZBEBMB  
                   |           |                               | • US West (Oregon)  
                   |           |                               | arn:aws:controltower:us-west-2::control/  
                   |           |                               | SQFKXDJCSXNS  
                   |           |                               | • Canada (Central)  
                   |           |                               | arn:aws:controltower:ca-central-1::control/  
                   |           |                               | BPAFLCPHRAROG  
                   |           |                               | • Asia Pacific (Sydney)  
                   |           |                               | arn:aws:controltower:ap-southeast-2::control/  
                   |           |                               | ZZTZVYPQVDAK  
                   |           |                               | • Asia Pacific (Singapore)  
                   |           |                               | arn:aws:controltower:ap-southeast-1::control/  
                   |           |                               | URWYQJCVIFQK  
                   |           |                               | • Europe (Frankfurt)  
                   |           |                               | arn:aws:controltower:eucentral-1::control/  
                   |           |                               | QCECBYGHSJFX  
                   |           |                               | • Europe (Ireland)  
                   |           |                               | arn:aws:controltower:eu-west-1::control/  
                   |           |                               | PEJTXAODKVKD  
                   |           |                               | • Europe (London)  
                   |           |                               | arn:aws:controltower:eu-west-2::control/  
                   |           |                               | EKJWCMKGILMM  
                   |           |                               | • Europe (Stockholm)  
                   |           |                               | arn:aws:controltower:eucentral-2::control/  
                   |           |                               | SQYCLPQYDQJH  
                   |           |                               | • Europe (North)  
                   |           |                               | arn:aws:controltower:eu-north-1::control/  
                   |           |                               | WDCPQYDQJH  
                   |           |                               | • Europe (West)  
                   |           |                               | arn:aws:controltower:eu-west-3::control/  
                   |           |                               | QECBYPQVDAK  
                   |           |                               | • Europe (South)  
                   |           |                               | arn:aws:controltower:eu-south-1::control/  
                   |           |                               | PEJTXAODKVKD  
                   |           |                               | • Europe (East)  
                   |           |                               | arn:aws:controltower:eu-east-1::control/  
                   |           |                               | QCECBYGHSJFX  
                   |           |                               | • Europe (South)  
                   |           |                               | arn:aws:controltower:eu-south-2::control/  
                   |           |                               | PEJTXAODKVKD  
                   |           |                               | • Europe (North)  
                   |           |                               | arn:aws:controltower:eu-north-2::control/  
                   |           |                               | WDCPQYDQJH  
                   |           |                               | • Asia Pacific (Osaka)  
                   |           |                               | arn:aws:controltower:ap-northeast-2::control/  
                   |           |                               | XQGJDGQQKGCRR  
                   |           |                               | • Asia Pacific (Taiwan)  
                   |           |                               | arn:aws:controltower:ap-southeast-1::control/  
                   |           |                               | URWYQJCVIFQK  
                   |           |                               | • Asia Pacific (Jakarta)  
                   |           |                               | arn:aws:controltower:ap-southeast-2::control/  
                   |           |                               | ZZTZVYPQVDAK  
                   |           |                               | • Asia Pacific (Tokyo)  
                   |           |                               | arn:aws:controltower:ap-northeast-1::control/  
                   |           |                               | XQGJDGQQKGCRR  |

2198
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.DynamoDB.2</td>
<td></td>
<td>Improve resiliency</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CP-10</td>
<td></td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/ITCPEFZTGH0G</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CP-9</td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/HKFSWAKSTQGF</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-12</td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/QKJWJTLZAEH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| • PCI DSS version 3.2.1 3.1 | central-1::control/LEFDCCDBLZGBP  
Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/SCPPKINMQZAF  
Asia Pacific (Singapore)  
ar:n:aws:controltower:ap-southeast-1::control/LVADITNDSYBI  
Europe (Frankfurt)  
ar:n:aws:controltower:eu-central-1::control/YSIW DenULVGH  
Europe (Ireland)  
ar:n:aws:controltower:eu-west-1::control/QJRPWNBCTHUF  
Europe (London)  ar:n:aws:controltower:eu-west-2::control/LAXIOZREATEL  
Europe (Stockholm)  
ar:n:aws:controltower:eu-north-1::control/JKHMWOIVIAEIY  
Asia Pacific (Mumbai)  
ar:n:aws:controltower:ap-south-1::control/LAWQMOXHJYFR  
Asia Pacific (Seoul)  
ar:n:aws:controltower:ap-northeast-2::control/EUJVQQYTHQHZ  
Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-northeast-1::control/ELRZAIXWJYGNK  
Europe (Paris)  
ar:n:aws:controltower:eu-west-3::control/URJSZZMBWZPK  
South America (São Paulo)  
ar:n:aws:controltower:sao-east-1::control/EVCIIVOS0HSML  
US West (N. California) |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.EC2.1           | • NIST 800-53 Rev 5 AC-21  
                    | • NIST 800-53 Rev 5 AC-3  
                    | • NIST 800-53 Rev 5 AC-3(7)  
                    | • NIST 800-53 Rev 5 AC-4  
                    | • NIST 800-53 Rev 5 AC-4(21)  
                    | • NIST 800-53 Rev 5 AC-6  
                    | • NIST 800-53 Rev 5 CA-9(1)  
                    | • NIST 800-53 Rev 5 CM-2  
                    | • NIST 800-53 Rev 5 CM-2(2)  
                    | • NIST 800-53 Rev 5 CM-7  
                    | • NIST 800-53 Rev 5 CM-8(1)  
                    | • NIST 800-53 Rev 5 SC-7  
                    | • NIST 800-53 Rev 5 SC-7(11)  
                    | • NIST 800-53 Rev 5 SC-7(16)  
                    | • NIST 800-53 Rev 5 SC-7(20)  
                    | • NIST 800-53 Rev 5 SC-7(21)  
                    | • NIST 800-53 Rev 5 SC-7(3)  
                    | • NIST 800-53 Rev 5 SC-7(4)  
                    | • NIST 800-53 Rev 5 SC-7(5)  
                    | • Enforce least privilege  
                    | • Limit network access | • US East (N. Virginia)  
                    |                     | arn:aws:controltower:us-east-1::control/  
                    |                     | FKQAQCYRILAK  
                    | • US East (Ohio)  
                    | arn:aws:controltower:us-east-2::control/  
                    | GFOMMGTLCZR  
                    | • US West (Oregon)  
                    | arn:aws:controltower:us-west-2::control/  
                    | TKXNMRFIMBEQ  
                    | • Canada (Central)  
                    | arn:aws:controltower:ca-central-1::control/  
                    | ZPHSVMVQOEUE  
                    | • Asia Pacific (Sydney)  
                    | arn:aws:controltower:ap-southeast-2::control/  
                    | QNCIIIGXDITVK  
                    | • Asia Pacific (Singapore)  
                    | arn:aws:controltower:ap-southeast-1::control/  
                    | BDCQIYFJGXwZ  
                    | • Europe (Frankfurt)  
                    | arn:aws:controltower:eu-central-1::control/  
                    | TTSVJOMQQLTN  
                    | • Europe (Ireland)  
                    | arn:aws:controltower:eu-west-1::control/  
                    | UBCGK0ORFCMO  
                    | • Europe (London)  
                    | arn:aws:controltower:eu-west-2::control/  
                    | XQBHIIQWMXMT  
                    | • Europe (Stockholm)  
                    | arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EC2.10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/VAUVRZKDKCKOY</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/TQGKBSI0QWYE</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/XZPDNCDQVUK</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SH.EC2.10

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EC2.10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-21</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/VAUVRZKDKCKOY</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/TQGKBSI0QWYE</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/XZPDNCDQVUK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>central-1::control/ YUTCRRLVNEDEL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ CMHFRDIISEEAB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ ZRINADVUBTHV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:europe-central-1::control/ PIVIIIALZUGQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:europe-west-1::control/ LQLAWRXQYGT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:europe-west-2::control/ VLMGUAIXJQHE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:europe-north-1::control/ IOVURZHPJU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ HBYPEQKUWCTB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ OXFAIHQZTPL</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ EPXMGYKIWO</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:europe-west-3::control/ IPUXYYDAQXVB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:south-1::control/ NSIDNZTQONXL</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-2::control/ YUTCRRLVNEDEL</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------</td>
<td>------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>SH.EC2.15</td>
<td>• NIST 800-53 Rev 5 AC-21</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/RJQGGVJZEBLT</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ULHSISJQFQEF</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/JSFKRLLFHAKSN</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/JNIOFIOSINQA</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/VSUCLWGVLQDF</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/FMFHQRLLYTUFU</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/DUHHRMNTZCW8</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/DJQXFSKNOMLU</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/YZCJTHFIDZSH</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-</td>
</tr>
</tbody>
</table>
### Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
SH.EC2.16 | • NIST 800-53 Rev 5 CM-8(1)  
• PCI DSS version 3.2.1 2.4 | • Protect configurations | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/LITUIAJFCNLG  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/QSBUZZBZQGLT  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/PBGUIXCOFNGC  
• Canada (Central)  
ar:n:aws:controltower:ca-2205
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>central-1::control/JOQQKGOKQTZN</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/EVNHPWACQHEA</td>
</tr>
<tr>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/DSMIAZVJBHYE</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/KXJKNJEDTXLK</td>
</tr>
<tr>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/HOTICOJPNKR</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/NRMUBCVGQGZX</td>
</tr>
<tr>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/UKRQRKXRAMKT</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/WHQJLTOONSGH</td>
</tr>
<tr>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/UNXYUHCHQWYQ</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/EHGVCOKGQGMI</td>
</tr>
<tr>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/DROYTLJVMLBR</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/SSWLJOJKAOLLL</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.EC2.17</td>
<td></td>
<td></td>
<td>us-east-1::control/</td>
</tr>
</tbody>
</table>

**SH.EC2.17**

- **Control identifier**: SH.EC2.17
- **Framework**: NIST 800-53 Rev 5 AC-4(21) and PCI DSS version 3.2.1 2.2
- **Control objective**: Limit network access
- **Control API identifiers, by Region**:
  - **US East (N. Virginia)**: arn:aws:controltower:us-east-1::control/ODNGXSKYGCAN
  - **US East (Ohio)**: arn:aws:controltower:us-east-2::control/GDSOIWYATGN
  - **US West (Oregon)**: arn:aws:controltower:us-west-2::control/DEEWSUBUCMPE
  - **Canada (Central)**: arn:aws:controltower:ca-central-1::control/HPCENFROXVMO
  - **Asia Pacific (Sydney)**: arn:aws:controltower:ap-southeast-2::control/LWYOJLPCBZPK
  - **Asia Pacific (Singapore)**: arn:aws:controltower:ap-southeast-1::control/HETFY0GBFWSA
  - **Europe (Frankfurt)**: arn:aws:controltower:eu-central-1::control/VRWUTNMJEXZW
  - **Europe (Ireland)**: arn:aws:controltower:eu-west-1::control/XGQPWTPUGBBB
  - **Europe (London)**: arn:aws:controltower:eu-west-2::control/HJJLVNYZLCKN
  - **Europe (Stockholm)**: arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EC2.18</td>
<td></td>
<td></td>
<td>north-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NFPERWALALZI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Japan (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WXFFLHKVJHMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ZQARMCQAP5SW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FAADSZJAEWQP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UELUSCFFWQBU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EC2.18</td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4(21)</td>
<td>Enforce least privilege</td>
<td>BKEEVLXJ01ZI</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td>VFZNFXKWHKOV</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td>USRKPFPWXBG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>central-1::control/KEIKZYHGWJS</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/UPVTHTZVSLWE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/DCMKUVKUGLNS</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/BTPGDLJBAMQH</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/VQQNYOUTJQT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/KZWNXDOXYBTQY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/KFLXJOLWUWRB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/CMOVKNTIMG0H</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/PULYBIGQAMIU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/WUKBMHWTCEY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/VDINNRDBMHL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/JRKGRIFEMDN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.EC2.19</td>
<td>• CIS AWS Benchmark 1.4 5.2</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/KTVMUABZKNOX</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/QUNSRZSYABZ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/QMYRIRVYDWC</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/FQPRTPPXKUX</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-1::control/SQJOUZGFZPV</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2(2)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/KEGOACYSSZL</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-7</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/BZIXCRFOIRCT</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/NRXPLVWHAEXY</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/FMQRFKULNUDA</td>
</tr>
</tbody>
</table>
|                    | • NIST 800-53 Rev 5 SC-7(16) |                    | • Europe (Stockholm) arn:aws:controltower:eu-
|                    | • NIST 800-53 Rev 5 SC-7(21) |                    | 2210
## SH.EC2.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EC2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CIS AWS Benchmark 1.4 5.1</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/CBXAVYGTSSXPR</td>
</tr>
<tr>
<td></td>
<td>• CIS AWS Benchmark 1.4 5.3</td>
<td>• Enforce least privilege</td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/TWIVVPDMFMYR</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-21</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/MVCYHCJCTKHQ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td></td>
<td>central-1::control/ARHTRDLRMOZS</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/RDJYZGSXFHYUE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/FXJGCOYWIZGN</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/GNOQGMZQNKDC</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2(2)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/TPTBSUKLVYNV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-7</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/YXDRBSXGSLSYA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-8(1)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/FWYHQCQTNVNT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-10</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/HRLSVBGX6GB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/YPREQUJHSSEG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-36</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/XKATTEDRNXMM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/TDDOHTFZXL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:saeast-1::control/SUPDKRQ3NJKG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
| SH.EC2.20 | • PCI DSS version 3.2.1 1.3.2  
• PCI DSS version 3.2.1 1.3.4  
• PCI DSS version 3.2.1 1.3.6  
• PCI DSS version 3.2.1 2.1  
• PCI DSS version 3.2.1 2.2  
• PCI DSS version 3.2.1 2.2.2  
• PCI DSS version 3.2.1 2.4 | • US West (N. California)  
arn:aws:controltower:us-west-1::control/FYKFHCWIPZEZ | |

### SH.EC2.20

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.EC2.20 | • NIST 800-53 Rev 5 CP-10  
• NIST 800-53 Rev 5 CP-6(2)  
• NIST 800-53 Rev 5 SC-36  
• NIST 800-53 Rev 5 SC-5(2)  
• NIST 800-53 Rev 5 SI-13(5)  
• PCI DSS version 3.2.1 1.2.1  
• PCI DSS version 3.2.1 1.3.2 | • Improve availability | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/SYKJWOLYPQA\G | • US East (Ohio)  
arn:aws:controltower:us-east-2::control/YXCGFXLUMVWD | • US West (Oregon)  
arn:aws:controltower:us-west-2::control/TCELFANHENO\F | • Canada (Central)  
arn:aws:controltower:ca-central-1::control/KLGN\NBLXGKP | • Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/XSSLFGGJDHRI | • Asia Pacific  
(Singapore)  
arn:aws:controltower:ap-southeast-1::control/IXQP\OKDTAS\VW | • Europe (Frankfurt)  
arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EC2.21</td>
<td></td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-west-1::control/HCJDMUGMARUR</td>
</tr>
</tbody>
</table>

SH.EC2.21

- CIS AWS Benchmark 1.4 5.1
- Limit network access
- US East (N. Virginia) arn:aws:controltower:us-west-1::control/HCJDMUGMARUR

- Europe (Ireland) arn:aws:controltower:eu-west-1::control/RNARDQCXVMPD
- Europe (London) arn:aws:controltower:eu-west-2::control/QKHXOCLKMPH
- Europe (Stockholm) arn:aws:controltower:eu-north-1::control/QQIFUCSRDDNN
- Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/WIULDPIAKWSU
- Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ODPVIBBCEMZV
- Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/DJMCSTIICRBQ
- Europe (Paris) arn:aws:controltower:eu-west-3::control/LAQNPJIXGRKN
- South America (São Paulo) arn:aws:controltower:sa-east-1::control/FVMQLYGBWWPP
- US West (N. California) arn:aws:controltower:us-west-1::control/HCJDMUGMARUR

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EC2.21</td>
<td></td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-west-1::control/HCJDMUGMARUR</td>
</tr>
</tbody>
</table>

SH.EC2.21

- CIS AWS Benchmark 1.4 5.1
- Limit network access
- US East (N. Virginia) arn:aws:controltower:us-west-1::control/HCJDMUGMARUR

Europe (Ireland) arn:aws:controltower:eu-west-1::control/RNARDQCXVMPD
- Europe (London) arn:aws:controltower:eu-west-2::control/QKHXOCLKMPH
- Europe (Stockholm) arn:aws:controltower:eu-north-1::control/QQIFUCSRDDNN
- Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/WIULDPIAKWSU
- Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ODPVIBBCEMZV
- Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/DJMCSTIICRBQ
- Europe (Paris) arn:aws:controltower:eu-west-3::control/LAQNPJIXGRKN
- South America (São Paulo) arn:aws:controltower:sa-east-1::control/FVMQLYGBWWPP
- US West (N. California) arn:aws:controltower:us-west-1::control/HCJDMUGMARUR
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>east-1::control/AVYCVZQFCQNU</td>
</tr>
<tr>
<td>• AC-4(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td>• CA-9(1)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/PCTQSZLBOZMZ</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td>• CM-2</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/DZFUOUZQSVYE</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td>• CM-2(2)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/KMVIVYSEGFWX</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td>• CM-7</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/YLOHXVLEFUBG</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>• SC-7</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/LQNNHNWZOCVA</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td>• SC-7(21)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/FLCQBKGZAFLW</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td>• SC-7(5)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/YMSHAPVGFZMR</td>
</tr>
<tr>
<td>PCI DSS version 3.2</td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td>• 1.2.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/WZMUKRGPNSNQX</td>
</tr>
<tr>
<td>PCI DSS version 3.2</td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td>• 1.3</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/ZAVXSARXXIWB</td>
</tr>
<tr>
<td>PCI DSS version 3.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>• 1.3.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/SOUTWZRNNWHMC</td>
</tr>
<tr>
<td>PCI DSS version 3.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>• 1.3.2</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/ZXUCIOCWKWESE</td>
</tr>
<tr>
<td>PCI DSS version 3.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>• 2.2</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------</td>
<td>--------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.EC2.22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 CM-8(1)</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td>arn:aws:controltower:us-east-1::control/ DKOGNVMOVXDM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PCI DSS version 3.2.1 2.4</td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Protect configurations</td>
<td>arn:aws:controltower:us-east-2::control/ ZIDBSURCQSXP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Limit network access</td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/ SPMPWQSGOLS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/ AHMNFXCLBAV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/ AJJFFKXXRAFMB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/ POVUMUFWGDAQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.EC2.25</td>
<td></td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/MEQBWNTEFWIE</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td></td>
<td>east-1::control/YCRFUQNWGOGU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **US East (Ohio)**
  arn:aws:controltower:us-east-2::control/KPVGGAIMGAT
- **US West (Oregon)**
  arn:aws:controltower:us-west-2::control/FYFOMHXXJRN
- **Canada (Central)**
  arn:aws:controltower:ca-central-1::control/GGJFODCQQCED
- **Asia Pacific (Sydney)**
  arn:aws:controltower:ap-southeast-2::control/QRHEHTVORUEM
- **Asia Pacific (Singapore)**
  arn:aws:controltower:ap-southeast-1::control/NSCQQMUEXIP
- **Europe (Frankfurt)**
  arn:aws:controltower:eu-central-1::control/VUVLCAOPMGSK
- **Europe (Ireland)**
  arn:aws:controltower:eu-west-1::control/LKSBRZYHPVM
- **Europe (London)**
  arn:aws:controltower:eu-west-2::control/KCCULCTGPQQU
- **Europe (Stockholm)**
  arn:aws:controltower:eu-north-1::control/EXZUBEASRXGZ
- **Asia Pacific (Mumbai)**
  arn:aws:controltower:ap-south-1::control/UXLJDHFLBJUT
- **Asia Pacific (Seoul)**
  arn:aws:controltower:ap-northeast-2::control/ZLQKQIGTIHZD
- **Asia Pacific (Tokyo)**
  arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EC2.3</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Encrypt data at rest</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/GPEXDJMATYQR</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/MAVIODWGWGFU</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/IEYBLJBRQTUY</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/QJPYFHYYITDF</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/CXPOISIRUAETB</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(6)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>central-1::control/OCODNWWJRA0</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/BXLYPFODSWXQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/TBZTSHAEMwV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/WSODWMGKOUOW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/WKHAUMVQJHBL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/SVKZ0QVUBAEW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/RKVSDKLNGUBT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/VDPJCLCQXXP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/TYBCYJHUUZHB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/OPFUDJAJEQG</td>
</tr>
</tbody>
</table>

**SH.EC2.4**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EC2.4</td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Optimize costs</td>
<td>US East (N. Virginia) arn:aws:controltower:us-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>east-1::control/ GTXSQEQJWO8FI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ CVCROPNPPPEC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ FICBNTRAXTL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ KWQVQHYVRNDF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ TWWDQASEKQF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ YPHZLBGOPEEF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ RJYQZKDGAPZ0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ EJHFSQMMOBQ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ MKJYAFFLOUTW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ SVYLMGWNLWDS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ FTPNARYEWDJYT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ QVKVKNHBRCV</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.EC2.6           | • CIS AWS Benchmark 1.4 3.9  
• NIST 800-53 Rev 5 AC-4(26)  
• NIST 800-53 Rev 5 AU-12  
• NIST 800-53 Rev 5 AU-2  
• NIST 800-53 Rev 5 AU-3  
• NIST 800-53 Rev 5 AU-6(3)  
• NIST 800-53 Rev 5 SI-7(8)  
• PCI DSS version 3.2.1 10.1  
• PCI DSS version 3.2.1 10.3.1  
• PCI DSS version 3.2.1 10.3.2 | • Establish logging and monitoring | northeast-1::control/ HXHUZBBSJPTTR  
• Europe (Paris) arn:aws:controltower:europe-west-3::control/ CUCAOWNGMHEU  
• South America (São Paulo) arn:aws:controltower:sa-east-1::control/ QOZOELBFFZTCX  
• US West (N. California) arn:aws:controltower:us-west-1::control/ VBBWMDZZYTJK  
• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ BEEYZGSRRAZM  
• US East (Ohio) arn:aws:controltower:us-east-2::control/ KGITVPNAWVRN  
• US West (Oregon) arn:aws:controltower:us-west-2::control/ PDPODRNNTZOK  
• Canada (Central) arn:aws:controltower:canada-central-1::control/ OKZLDNISDVIX  
• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ NZTMIPSZHGMV  
• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ C0KPYBQATQFC  
• Europe (Frankfurt) arn:aws:controltower:eu
### SH.EC2.7

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EC2.7</td>
<td>• CIS AWS Benchmark 1.4 2.2.1</td>
<td>• Encrypt data at rest</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-west-1::control/ UJGZYPQJITPC</td>
</tr>
</tbody>
</table>

- **Framework**
  - • PCI DSS version 3.2.1 10.3.3
  - • PCI DSS version 3.2.1 10.3.4
  - • PCI DSS version 3.2.1 10.3.5
  - • PCI DSS version 3.2.1 10.3.6
  - • PCI DSS version 3.2.1 2.2

- **Control objective**
  - central-1::control/FWYAUKSIVHLF
  - • Europe (Ireland) arn:aws:controltower:eu-west-1::control/EKEHTOJRBDTQ
  - • Europe (London) arn:aws:controltower:eu-west-2::control/CTTUXWTGFDAQ
  - • Europe (Stockholm) arn:aws:controltower:eu-north-1::control/XMDSIZZAWLKR
  - • Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/XSSFYRGJ2NXH
  - • Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/GIBTSEIVFPKI
  - • Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/OTTUNOMCDOZ
  - • Europe (Paris) arn:aws:controltower:eu-west-3::control/FWXUOKAJABW
  - • South America (São Paulo) arn:aws:controltower:sa-east-1::control/VBVVYZQYOWSO
  - • US West (N. California) arn:aws:controltower:us-west-1::control/UJGZYPQJITPC
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td></td>
<td>east-1::control/AIQQAEVYZWFE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/OVKQLAORAIIBS</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/JONOVBLPBWAW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/XGNQTADZLIOK</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/KOPZMCROGHHM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ROKMCRMRLKWF</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/NYJHLLLGLJNVL</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ILUNFEJNBVLU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/FHAGCBGOFCZO</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/KXOHPXWPWJQZ</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/YGQRTWXYLOX</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/PPUMEARUOMA5</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-</td>
</tr>
</tbody>
</table>
### SH.EC2.8

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EC2.8</td>
<td></td>
<td>Protect configurations</td>
<td>northeas-1::control/ AMMZARWOCRGP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) ark:aws:controltower:eu-west-3::control/KKPRSSACQYWB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (Sao Paulo) ark:aws:controltower:sa-east-1::control/DQPCEGISFCNR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California) ark:aws:controltower:us-west-1::control/MFFNOSWTXLHJ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td>Protect configurations</td>
<td>US East (N. Virginia) ark:aws:controltower:us-east-1::control/RZXGVSCOVETI</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(15)</td>
<td>Protect configurations</td>
<td>US East (Ohio) ark:aws:controltower:us-east-2::control/LUMAXIDLGBKZV</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td>Protect configurations</td>
<td>US West (Oregon) ark:aws:controltower:us-west-2::control/IFGERUTZUTFR</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.1.1</td>
<td>Protect configurations</td>
<td>Canada (Central) ark:aws:controltower:ca-central-1::control/HXUBJTXHYPI</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.2.1</td>
<td>Protect configurations</td>
<td>Asia Pacific (Sydney) ark:aws:controltower:ap-southeast-2::control/JIQJXDPPNWCG</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.2.2</td>
<td>Protect configurations</td>
<td>Asia Pacific (Singapore) ark:aws:controltower:ap-southeast-1::control/PJIPLFCSHLGZ</td>
</tr>
<tr>
<td></td>
<td>Europe (Frankfurt) ark:aws:controltower:eu-</td>
<td>Protect configurations</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.EC2.9</td>
<td>NIST 800-53 Rev 5 AC-21</td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-west-1::control/DUWQHNJEqKNS</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td></td>
<td>east-1::control/MINKVOGJPJARR</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/RYXXK6SIEIVR</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/KXUMGWLD2GDKW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/XPPJCAOJDJFWB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/MQLRBDKQRBZ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/WHRCL3JDRMG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/DWBLDYSTIZXJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/GAVOL3J3LJLI</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/JQNVRYPJWYUEUN</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/GDLWNHNBZSC</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/DJYJFJKFZGB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/EXUUFYGYJTSR</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.ECR.1</td>
<td></td>
<td>Manage vulnerabilities</td>
<td></td>
</tr>
</tbody>
</table>

- **NIST 800-53 Rev 5 RA-5**
- **PCI DSS version 3.2.1 11.2.3**
- **PCI DSS version 3.2.1 6.3.2**

- **US East (N. Virginia)**
  arn:aws:controltower:us-east-1::control/ OZQOTZVQZXUY

- **US East (Ohio)**
  arn:aws:controltower:us-east-2::control/ JQCJORMCEEUT

- **US West (Oregon)**
  arn:aws:controltower:us-west-2::control/ LIWRSKLRMDEW

- **Canada (Central)**
  arn:aws:controltower:ca-central-1::control/ VJWDDYTOKMDLE

- **Asia Pacific (Sydney)**
  arn:aws:controltower:ap-southeast-2::control/ KTMZ00LGTWJX

- **Asia Pacific (Singapore)**
  arn:aws:controltower:ap-southeast-1::control/ DISETFBTTHBO

- **Europe (Frankfurt)**
  arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ECR.2</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Protect configurations</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ JEZPGFQEPEQL</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ JWNRPDFKGDYD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-8(1)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ ICDSCZRMSPAGC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ SKIMJMSMPJYN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.4</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ THACMOMUCYAC</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ TKDSBFRKCMRHM</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ WNRNZLQUPYQ</td>
</tr>
<tr>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ MTSVJIZJRHG</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ DOWLYNHVRUPZ</td>
</tr>
<tr>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ FMXGUCETFMHU</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ LOYORBYGCYNU</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ OCKPMLOSKZGW |          |                  | Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ECR.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SH.ECR.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td>• Manage vulnerabilities</td>
<td>• US East (N. Virginia)</td>
<td>northeast-1::control/UDEEBKULXVSOR</td>
</tr>
<tr>
<td>CA-9(1)</td>
<td>• Protect configurations</td>
<td>arn:aws:controltower:us-east-1::control/UDDDYEJAYHXP</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td>• PCI DSS version 3.2.1</td>
<td>• US East (Ohio)</td>
<td>northeast-1::control/CFENKYCNLFUQ</td>
</tr>
<tr>
<td>CM-2</td>
<td>3.1</td>
<td>arn:aws:controltower:us-east-2::control/PQBPPQRIRVDR</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td>• Manage vulnerabilities</td>
<td>• US West (Oregon)</td>
<td>northeast-1::control/HKFGMXYZUIZU</td>
</tr>
<tr>
<td>CM-2(2)</td>
<td>• Protect configurations</td>
<td>arn:aws:controltower:us-west-2::control/HDESRSEEGACA</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>• Manage vulnerabilities</td>
<td>• Canada (Central)</td>
<td>northeast-1::control/UUKJMLKQDJSZ</td>
</tr>
<tr>
<td>3.1</td>
<td>• Protect configurations</td>
<td>arn:aws:controltower:ap-southeast-2::control/VMXVKJDQHDKB</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>• Manage vulnerabilities</td>
<td>• Asia Pacific (Sydney)</td>
<td>northeast-1::control/XOFCMMRFEKZ</td>
</tr>
<tr>
<td>3.1</td>
<td>• Protect configurations</td>
<td>arn:aws:controltower:ap-southeast-1::control/XOFCMMRFEKZ</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td>• Manage vulnerabilities</td>
<td>• Asia Pacific (Singapore)</td>
<td>northeast-1::control/HKFGMXYZUIZU</td>
</tr>
</tbody>
</table>
| 3.1                | • Protect configurations | arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ECS.1</td>
<td>NIST 800-53 Rev 5 AC-2(1)</td>
<td>Manage vulnerabilities</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ PKBHCVZWSUM</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------------</td>
<td>-------------------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>east-1::control/OPQLRHGSJ3JBU</td>
</tr>
<tr>
<td>AC-3</td>
<td></td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/YQDCIBD0UYPTD</td>
</tr>
<tr>
<td>AC-3(15)</td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/PZJZHGWZUIMB</td>
</tr>
<tr>
<td>AC-3(7)</td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/ZBKHVVFKVDWR</td>
</tr>
<tr>
<td>AU-6(3)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/CPDDHHDPDKIP</td>
</tr>
<tr>
<td>AU-6(4)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/DRQQFHI0IXFD</td>
</tr>
<tr>
<td>CA-7</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/XDQJHTGYRMXH</td>
</tr>
<tr>
<td>SI-2</td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/JUUTNCMUDPKT</td>
</tr>
<tr>
<td>SI-2(2)</td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/WVYXIBWPTVGT</td>
</tr>
<tr>
<td>SI-2(4)</td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/VPIMQOUEKYAC</td>
</tr>
<tr>
<td>SI-2(5)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/DZKNSGRUQHTT</td>
</tr>
<tr>
<td>10.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/CRTKLEJMVWXK</td>
</tr>
<tr>
<td>10.2.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
</tbody>
</table>
### SH.ECS.10

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.ECS.10          | NIST 800-53 Rev 5 SI-2         | Manage vulnerabilities      | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/WWVYZKSEXPPK  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/ABBRGFOMERCE  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/LOZWHACNQRQQ  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/ULWMHFCYLPWR  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/WILNHTOPFEWQ  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/JLJTQCESWDDA  
• Europe (Frankfurt)  
arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ECS.12</td>
<td>NIST 800-53 Rev 5 AU-6(3)</td>
<td>Establish logging and monitoring</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ANFBKAZJRLAL</td>
</tr>
</tbody>
</table>

- Europe (Ireland) arn:aws:controltower:eu-west-1::control/TAHWMCHCPTII
- Europe (London) arn:aws:controltower:eu-west-2::control/WTDCOBPXKBVY
- Europe (Stockholm) arn:aws:controltower:eu-north-1::control/EBJTJOMBWLQ
- Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/SUDPRJEDUAWI
- Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/MFBANBYEIBII
- Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ALVOGJRVKKZK
- Europe (Paris) arn:aws:controltower:eu-west-3::control/XUEPVHSVPADK
- South America (São Paulo) arn:aws:controltower:sa-east-1::control/NLUIVMHMGHOA
- US West (N. California) arn:aws:controltower:us-west-1::control/ANFBKAZJRLAL
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST 800-53 Rev 5 AU-6(4)</td>
<td>•</td>
<td></td>
<td>• east-1::control/LWIVFPXZOSHV</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 CA-7</td>
<td>•</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/FZX0IKGO0NZN</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SI-2</td>
<td>•</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/TPAZNVVAIDJA</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.1</td>
<td>•</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/KKERYZUCG6GJC</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.2.1</td>
<td>•</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/MGJPDOBPUTIT</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.2.2</td>
<td>•</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/EVNJGVRMBUIZ</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.2.3</td>
<td>•</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/TTHICLWSSHFL</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.2.4</td>
<td>•</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/NCQMUQIVVXCS</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.2.5</td>
<td>•</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/GIMEJGBOEQRI</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.2.7</td>
<td>•</td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/UOGMDFBVJPQJ</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.3.1</td>
<td>•</td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/XAFJJVQHMGNX</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.3.2</td>
<td>•</td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/QILRKMIJRFVDI</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 10.3.3</td>
<td>•</td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.ECS.2</td>
<td></td>
<td></td>
<td>northeast-1::control/EOYYFDCHIFPM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/PQEJONXBITPM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(São Paulo) arn:aws:controltower:sa-east-1::control/EXXSLRJHTNEZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/PJKCMWEKLDRL</td>
</tr>
</tbody>
</table>

**SH.ECS.2**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ECS.2</td>
<td>• NIST 800-53 Rev 5 AC-21</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td>• Enforce least privilege</td>
<td>arn:aws:controltower:us-east-1::control/JCFLSLPRFWPS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/HGHFAHJTOSSD</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/ITRPZWOLQECJ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/BEDXPJRWTDZQ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/OAVLGBTROIFK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/ZPDCBKMMLWIIO</td>
</tr>
<tr>
<td></td>
<td>• Europe (Frankfurt)</td>
<td></td>
<td>arn:aws:controltower:eu-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>central-1::control/JVNVFXHHI0DF</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:euc-west-1::control/HCEOCFPEDPJB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:euc-west-2::control/MGZAPMBPTJJC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td>Protect configurations</td>
<td>Europe (Stockholm) arn:aws:controltower:euc-north-1::control/GEWFDODMQPJO</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/OCQSRQ0SDJOY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/AZEUZHANYCTW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/DRVAMWQPGK</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td>• Enforce least privilege</td>
<td>east-1::control/UFYTWRSCKNKN</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.7</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/DESMSSIJBYWH</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/KYMDJLKLXJHG</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/SFBMOXOHPGJA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/XTGMINJOWXW</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/CQYNJHWPJMJZ</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/SNWDOPIOUEOV</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/XKOKVXODCBPP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/WWVNINIGVLQY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/KNUALLKQETTY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ZMMJYDDXRGGF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/VJYBXQIPBUUX</td>
<td></td>
</tr>
</tbody>
</table>
| | | • Asia Pacific (Tokyo) arn:aws:controltower:ap-
## SH.ECS.4

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.ECS.4            | • NIST 800-53 Rev 5 AC-2(1)  
                     • NIST 800-53 Rev 5 AC-3  
                     • NIST 800-53 Rev 5 AC-3(15)  
                     • NIST 800-53 Rev 5 AC-3(7)  
                     • NIST 800-53 Rev 5 AC-5  
                     • NIST 800-53 Rev 5 AC-6  
                     • PCI DSS version 3.2.1 7.1.1  
                     • PCI DSS version 3.2.1 7.2.1  
                     • PCI DSS version 3.2.1 7.2.2 | • Enforce least privilege | • US East (N. Virginia)  
                               arn:aws:controltower:us-east-1::control/  
                               ZTDLFKNABSTB  
                               • US East (Ohio)  
                               arn:aws:controltower:us-east-2::control/  
                               OCQLRQRWYOGE  
                               • US West (Oregon)  
                               arn:aws:controltower:us-west-2::control/  
                               MRRRZFTNEETG  
                               • Canada (Central)  
                               arn:aws:controltower:ca-central-1::control/  
                               IFCCVDCGNNTK  
                               • Asia Pacific (Sydney)  
                               arn:aws:controltower:ap-southeast-2::control/  
                               LFUVYWYFYYLU  
                               • Asia Pacific (Singapore)  
                               arn:aws:controltower:ap-southeast-1::control/  
                               WYHAZMRJHMRS  
                               • Europe (Frankfurt)  
                               arn:aws:controltower:eu- |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ECS.5</td>
<td>NIST 800-53 Rev 5 AC-2(1)</td>
<td>Enforce least privilege</td>
<td>US East (N. Virginia) arn:aws:controltower:us-west-1::control/</td>
</tr>
</tbody>
</table>

**SH.ECS.5**
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td>• Protect configurations</td>
<td>east-1::control/KPIETXSDELVX</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/DZURSFLDZJCZ</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/GTHMXNOZDNXN</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-5</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/XGGLVMQEYMTU</td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/XWHFZMHURMPA</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/GETZOYOUJIXQ</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/JRZBGRQCXVUK</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/UIKDFBESIRCQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/PAQMTHTZYVV</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ZUITZQVJNUJ</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/AFNCBEGBYIXM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/RXTACLFBSCY</td>
<td></td>
</tr>
</tbody>
</table>
| | | • Asia Pacific (Tokyo) arn:aws:controltower:ap-
### Control API identifiers, by Region

<table>
<thead>
<tr>
<th>Framework</th>
<th>Control objective</th>
<th>Control identifier</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST 800-53  Rev 5 CA-9(1)</td>
<td>Use strong authentication</td>
<td>SH.ECS.8</td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td>NIST 800-53  Rev 5 CM-2</td>
<td></td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td>Canada (Central)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
</tbody>
</table>

- **US East (N. Virginia)**
  - arn:aws:controltower:us-east-1::control/LJQVFMBPQAQV
- **US East (Ohio)**
  - arn:aws:controltower:us-east-2::control/NAWGNOLLBSXU
- **US West (Oregon)**
  - arn:aws:controltower:us-west-2::control/VHVNQSXRDIET
- **Canada (Central)**
  - arn:aws:controltower:ca-central-1::control/OTWMMXXNMCSP
- **Asia Pacific (Sydney)**
  - arn:aws:controltower:ap-southeast-2::control/HLGKEFJGGOB
- **Asia Pacific (Singapore)**
  - arn:aws:controltower:ap-southeast-1::control/WQZEOJHIDDT
- **Europe (Frankfurt)**
  - arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EFS.1</td>
<td></td>
<td>Encrypt data at rest</td>
<td>central-1::control/RTATKKXPWIRIB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:europe-west-1::control/EMXDVHIJLRSJH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:europe-west-2::control/BPVJCGDDGSQY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:europe-north-1::control/HGLDPNFOQMBH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:asia-pacific::control/KANMMZEMJFTV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:asia-pacific-northeast-2::control/GNPKSCS2TRDA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:asia-pacific-northeast-1::control/FXUBXSUJWJP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:europe-west-3::control/MVPNPMJPAIV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:south america::control/RNZTOFGURZUK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/UBMPZYX0EOSM</td>
</tr>
</tbody>
</table>

**SH.EFS.1**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EFS.1</td>
<td></td>
<td>Encrypt data at rest</td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (N. Virginia) arn:aws:controltower:us-west-1::control/UBMPZYX0EOSM</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td></td>
<td>east-1::control/DAAY0ZMUFPOA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/TUBTOQPYRFCC</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/SSNRPFOJAFFZ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/CSVENQNTZNYV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/TTLYUNPAEQS</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/FCPQPNIVMTLV</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/EEWVZ50ALXSA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/YRPBNRJZQEH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/BNXMOCOJDLX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/QNVQPFIVIXEA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/IUAGTQHNNXJJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/YNHULGHOVNE</td>
</tr>
</tbody>
</table>
| | | | • Asia Pacific (Tokyo) arn:aws:controltower:ap-
### SH.EFS.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.EFS.2           |           | Improve resiliency | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/AKKPASMFZEGR  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/PDXZZWIPVJJY  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/PGSGUFPZMDK  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/GCWIQEDCGVFE  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/OJAGQJKAXRQT  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/LNAWKNUYDXZM  
• Europe (Frankfurt)  
arn:aws:controltower:eu-west-1::control/DOBFVCUKROMU  
• Europe (Paris)  
arn:aws:controltower:eu-west-3::control/TPNLRFVPUQON  
• South America (São Paulo)  
arn:aws:controltower:sa-east-1::control/GJQNS3LZRNTD  
• US West (N. California)  
arn:aws:controltower:us-west-1::control/ZOQSLEMJOPAG  

- NIST 800-53 Rev 5 CP-10  
- NIST 800-53 Rev 5 CP-6  
- NIST 800-53 Rev 5 CP-6(1)  
- NIST 800-53 Rev 5 CP-6(2)  
- NIST 800-53 Rev 5 CP-9  
- NIST 800-53 Rev 5 SC-5(2)  
- NIST 800-53 Rev 5 SI-12  
- NIST 800-53 Rev 5 SI-13(5)  
- PCI DSS version 3.2.1 3.1
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EFS.3</td>
<td></td>
<td>NIST 800-53 Rev 5 AC-6(10)</td>
<td>US East (N. Virginia) arn:aws:controltower:us-west-1::control/YNYOCEJWWSRB</td>
</tr>
</tbody>
</table>

**SH.EFS.3**

- **Europe (Ireland)**
  - arn:aws:controltower:eu-west-1::control/QRUYZOHTBYFY
- **Europe (London)**
  - arn:aws:controltower:eu-west-2::control/CRQDNORELCAL
- **Europe (Stockholm)**
  - arn:aws:controltower:eu-north-1::control/FGMJDIKXKHPE
- **Asia Pacific (Mumbai)**
  - arn:aws:controltower:ap-south-1::control/RYDLPNLERROW
- **Asia Pacific (Seoul)**
  - arn:aws:controltower:ap-northeast-2::control/VTEXLEQANDAO
- **Asia Pacific (Tokyo)**
  - arn:aws:controltower:ap-northeast-1::control/POMNJTTHRCLYE
- **Europe (Paris)**
  - arn:aws:controltower:eu-west-3::control/YVVVIWNYDVW
- **South America (São Paulo)**
  - arn:aws:controltower:sa-east-1::control/PTOQRJVREEMJ
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td></td>
<td>east-1::control/ YLSKXEGSHZFO</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ DZQNOWWCJPFI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ NVESMCWZHHTI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ JJUVJECZICXY</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ MLSTPREPKCET</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-2::control/ OPPEWLVDKCKA</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-1::control/ MLSTPREPKCET</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ ZVUGCKESDOAV</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ ZVUGCKESDOAV</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ FWIBVWJMMMAI</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ RIRPTGIVTUUG</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ QZXMNOHKTGH</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ FMAXTHSWUHAN</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo) | | | Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EFS.4</td>
<td>• NIST 800-53 Rev 5 AC-6(2)</td>
<td>• Enforce least privilege</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/IINULENBTWXX</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/NLBIOAOKHRRAW</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/SMRVHSXRVJWI</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/QSSJLANZTBBW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/UWYRRPBPEEVO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/VBQLOLHZXXNI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-</td>
</tr>
</tbody>
</table>

SH.EFS.4
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.EKS.2</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Manage vulnerabilities</td>
<td>US East (N. Virginia) &lt;br&gt;arn:aws:controltower:us-east-1::control/OXIESZJRCGZM</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
<td>east-1::control/</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td></td>
<td>RDAUUJAGICFK</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SI-2(2)</td>
<td></td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SI-2(4)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SI-2(5)</td>
<td></td>
<td></td>
<td>ZFJKLNNVBGRM</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 6.2</td>
<td></td>
<td></td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KTILUSDPMIXF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NXDAILOKJURD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DAMQGUZUZBH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ZGYTSFAJVCJCI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>QXPCMQZGPBRZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>YSQVVFCPJXKE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>YIJVYZLMAWXZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NNTRKKQFPKQKB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OLRVJ3BCRSGN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KNMBERLUNBLY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.ELB.10          | • NIST 800-53 Rev 5 CP-10  
|                    | • NIST 800-53 Rev 5 CP-6(2)  
|                    | • NIST 800-53 Rev 5 SC-36  
|                    | • NIST 800-53 Rev 5 SC-5(2)  
|                    | • NIST 800-53 Rev 5 SI-13(5)  | • Improve availability | • US East (N. Virginia)  
|                    |                      |                                | arn:aws:controltower:us-east-1::control/  
|                    |                      |                                | FVCVSQHZTIZT  
|                    |                      |                                | • US East (Ohio)  
|                    |                      |                                | arn:aws:controltower:us-east-2::control/  
|                    |                      |                                | GPJSWKIEBWUR  
|                    |                      |                                | • US West (Oregon)  
|                    |                      |                                | arn:aws:controltower:us-west-2::control/  
|                    |                      |                                | USWBDHCGRVRDW  
|                    |                      |                                | • Canada (Central)  
|                    |                      |                                | arn:aws:controltower:ca-central-1::control/  
|                    |                      |                                | RVBQKRTS1QHB  
|                    |                      |                                | • Asia Pacific (Sydney)  
|                    |                      |                                | arn:aws:controltower:ap-southeast-2::control/  
|                    |                      |                                | ZDVYGBSRJWYG  
|                    |                      |                                | • Asia Pacific (Singapore)  
|                    |                      |                                | arn:aws:controltower:ap-southeast-1::control/  
|                    |                      |                                | FJZFXVJRPRAZ  
|                    |                      |                                | • Europe (Frankfurt)  
|                    |                      |                                | arn:aws:controltower:eu- |
### SH.ELB.12

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.ELB.12          |           | Protect data integrity | US East (N. Virginia)  
arn:aws:controltower:us-west-1::control/ WSUSFKNXHPIX |

- NIST 800-53 Rev 5  
AC-4(21)

- Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/ KXDJVFVEKKB

- Europe (London)  
arn:aws:controltower:eu-west-2::control/ FAOBZSYTLPT

- Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/ XTYTRYANVABU

- Asia Pacific (Mumbai)  
arn:aws:controltower:ap-south-1::control/ SNUVEKZNRWV

- Asia Pacific (Seoul)  
arn:aws:controltower:ap-northeast-2::control/ KXJZKDGJXTYM

- Asia Pacific (Tokyo)  
arn:aws:controltower:ap-northeast-1::control/ QCGDTTLRIYXP

- Europe (Paris)  
arn:aws:controltower:eu-west-3::control/ XBJIKRKMXAO

- South America (São Paulo)  
arn:aws:controltower:sa-east-1::control/ WDXWFRIIXIUZ

- US West (N. California)  
arn:aws:controltower:us-west-1::control/ WSUSFKNXHPIX
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>east-1::control/ ZLQPMXHBILJA</td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ VGZVYNURBZDT</td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.2.1</td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ TUKTIBCDJAVB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.3</td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ UXUMUBGBZH2O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.3.1</td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ HAQAZMDWOYQJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.3.2</td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ VDPGRRYBGHOV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 1.3.4</td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ VPPWCERAPGT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 2.2.3</td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ QVTSOBDYQEAJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1 6.6</td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ JYQSYKANCTJW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ LKVZZNIHLSAG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ QXHRCRDLPKYU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-south-2::control/ LKVZZNIHLSAG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.ELB.14</td>
<td></td>
<td>Protect data integrity</td>
<td>🌐 IHSKURWXDUTW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protect data integrity</td>
<td>🌐 Europe (Ireland) arn:aws:controltower:eu-west-1::control/ XLTLBTQHXLX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protect data integrity</td>
<td>🌐 Europe (London) arn:aws:controltower:eu-west-2::control/ QSDQGJCOPYS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protect data integrity</td>
<td>🌐 Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ SRJYXMTWSTJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protect data integrity</td>
<td>🌐 Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ WMXUZBGXGQW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protect data integrity</td>
<td>🌐 Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ RAMKNUUEEXG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protect data integrity</td>
<td>🌐 Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ WARGNXCHVUV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protect data integrity</td>
<td>🌐 Europe (Paris) arn:aws:controltower:eu-west-3::control/ PIVSOAXMRWDV</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protect data integrity</td>
<td>🌐 South America (São Paulo) arn:aws:controltower:sa-east-1::control/ PKDINMVJGHFK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protect data integrity</td>
<td>🌐 US West (N. California) arn:aws:controltower:us-west-1::control/ BXDUORERQDGQ</td>
</tr>
</tbody>
</table>

---

**SH.ELB.14**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ELB.14</td>
<td></td>
<td>Protect data integrity</td>
<td>🌐 US East (N. Virginia) arn:aws:controltower:us-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td></td>
<td>east-1::control/YFWKLA9FUB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/WCWDSWOQVL VAX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/DQHULQCNCE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/HABOKKLHBFX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/NTKIROXXRLDD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/WUXJUZ0QMNC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/EWPWYQNIWE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.3</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/HUBSUIYPXEUM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 6.6</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/JHDFVZTRLQDD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/KUDRC OMEIPTR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/GFCILRDCWFSU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/JMOCSNTDNPP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-</td>
</tr>
</tbody>
</table>
### Control identifier | Framework | Control objective | Control API identifiers, by Region
---|---|---|---
| SH.ELB.2 | NIST 800-53 Rev 5 AC-17(2) | Encrypt data in transit | US East (N. Virginia) arn:aws:controltower:us-east-1::control/CDLXUWUBOOTY
| | NIST 800-53 Rev 5 AC-4 | | US East (Ohio) arn:aws:controltower:us-east-2::control/GIDGHLCTSHTW
| | NIST 800-53 Rev 5 IA-5(1) | | US West (Oregon) arn:aws:controltower:us-west-2::control/SBRXFCDRVVFI
| | NIST 800-53 Rev 5 SC-12(3) | | Canada (Central) arn:aws:controltower:ca-central-1::control/GWNPDCMTNXYF
| | NIST 800-53 Rev 5 SC-13 | | Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/HQUTVBJZPCUEF
| | NIST 800-53 Rev 5 SC-23 | | Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/FNNZMAARRNKG
| | NIST 800-53 Rev 5 SC-23(5) | | Europe (Frankfurt) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ELB.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3 4.1 8.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-17(2)</td>
<td></td>
<td>Encrypt data in transit</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-west-1::control/USRINZFLQJTX</td>
</tr>
</tbody>
</table>

AWS Control Tower User Guide
SH.ELB.3
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td>AC-4</td>
<td>east-1::control/</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td>IA-5(1)</td>
<td>FWOYTUWHEYQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td>SC-12(3)</td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td>SC-13</td>
<td>arn:aws:controltower:us-east-2::control/</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td>SC-23</td>
<td>YWXMTFLLBGJP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td>SC-23(3)</td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(4)</td>
<td>arn:aws:controltower:us-west-2::control/</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td>SC-8</td>
<td>DFCYTONTBHIGM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td>SC-8(1)</td>
<td>Canada (Central)</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5</td>
<td>SC-8(2)</td>
<td>arn:aws:controltower:ca-central-1::control/</td>
</tr>
<tr>
<td></td>
<td>SC-7(6)</td>
<td>SI-7(6)</td>
<td>GVQODFJYLZWC</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1</td>
<td>2.3</td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1</td>
<td>4.1</td>
<td>arn:aws:controltower:ap-southeast-2::control/</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1</td>
<td>8.2.1</td>
<td>WLNVBASSPYBO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BG6EOIEXPOID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BKTUULNOSCAW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MBTVIFQQQMUF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PMRUICASDLHV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>YESXDSARDRYS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KTZTXRNIMGNCY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RHKBQSPWEASW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
</tr>
</tbody>
</table>
|                         |                     |                   | arn:aws:controltower:ap-
Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
SH.ELB.4 | • NIST 800-53 Rev 5 SC-7(4)
• NIST 800-53 Rev 5 SC-8(2)
• PCI DSS version 3.2.1 1.2.1
• PCI DSS version 3.2.1 1.3
• PCI DSS version 3.2.1 1.3.1
• PCI DSS version 3.2.1 1.3.2
• PCI DSS version 3.2.1 1.3.4 | • Protect configurations | • US East (N. Virginia)
arn:aws:controltower:us-east-1::control/ HBTZGRURMOKM
• US East (Ohio)
arnc:aws:controltower:us-east-2::control/ ROBASCYMSHHJ
• US West (Oregon)
arnc:aws:controltower:us-west-2::control/ OELFZGUMNKN
• Canada (Central)
arnc:aws:controltower:ca-central-1::control/ EPRDRSPZQGPS
• Asia Pacific (Sydney)
arnc:aws:controltower:ap-southeast-2::control/ FIPLHETXVCE
• Asia Pacific (Singapore)
arnc:aws:controltower:ap-southeast-1::control/ TGUHGXFNQXKW
• Europe (Frankfurt)
arnc:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ELB.5</td>
<td>NIST 800-53 Rev 5</td>
<td>Establish logging and monitoring</td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>AC-4(26)</td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>XINTTRUSSMYM</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-10</td>
<td>east-1::control/RRDKKWWTNZOH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US East (Ohio)</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/VURKNNKDBOFS</td>
<td></td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/VVOADNJLLYG5</td>
<td></td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:central-1::control/FBADZSAQTQEX</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/HPXOCKUZZZONE</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/NKENZBRJPOZ</td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/CQGLVPQHETRU</td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ZSCPMMFXXTQY</td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/XTOUKDNICGIV</td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ZSBQEDXDBMGN</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/FOTLYKSQLKLV</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/IVVPLPKPUUWQ</td>
<td></td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo) | | Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.ELB.6           |           | Improve availability | US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/SYVZPNNPLEGU  
US East (Ohio)  
arn:aws:controltower:us-east-2::control/GXDETIMMTQLK  
US West (Oregon)  
arn:aws:controltower:us-west-2::control/HVEKMNKLWBH  
Canada (Central)  
arn:aws:controltower:ca-central-1::control/UBKSMZHRBUQV  
Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/PJAZYXFFMUVW  
Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/BDRMHLFBGUEP  
Europe (Frankfurt)  
arn:aws:controltower:eu-

### SH.ELB.6

- NIST 800-53 Rev 5 CA-9(1)
- NIST 800-53 Rev 5 CM-2
- NIST 800-53 Rev 5 CM-2(2)
- NIST 800-53 Rev 5 CM-3
- NIST 800-53 Rev 5 SC-5(2)
- PCI DSS version 3.2.1 2.2

### northeast-1::control/OXVTPFBQEEAQ
- Europe (Paris)  
arwn:aws:controltower:eu-west-3::control/NJNQRVLHTOEL
- South America (São Paulo)  
arwn:aws:controltower:sa-east-1::control/EYYHLNPZCJJV
- US West (N. California)  
arwn:aws:controltower:us-west-1::control/BCKE0NH5CFNJ

```
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ELB.7</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Improve resiliency</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ EWW5SQVZDTHVM</td>
</tr>
</tbody>
</table>

**SH.ELB.7**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ELB.7</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Improve resiliency</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ EWW5SQVZDTHVM</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
<td>east-1::control/WHVUNZVVRTS Q</td>
</tr>
<tr>
<td>• US East (Ohio)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/NY0ZBRSAFFEV</td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/HXVRPCCCRZ YY</td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/RRQTKRFGJHJ IU</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/LPNJZVNGKYJN</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/FIASTWTNRWVC</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/MASPMNHSXITG</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/THSXODYFVSSS</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/VEJYGWLULHGK</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/SHYPARLBHAML</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/IKKHFUAQZUVN</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/NKYEWHLZPPQC</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo) | | | arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.ELB.8           | • NIST 800-53 Rev 5 AC-17(2)  
                        • NIST 800-53 Rev 5 AC-4  
                        • NIST 800-53 Rev 5 IA-5(1)  
                        • NIST 800-53 Rev 5 SC-12(3)  
                        • NIST 800-53 Rev 5 SC-13  
                        • NIST 800-53 Rev 5 SC-23  
                        • NIST 800-53 Rev 5 SC-23(3)  
                        • NIST 800-53 Rev 5 SC-7(4)  
                        • NIST 800-53 Rev 5 SC-8  
                        • NIST 800-53 Rev 5 SC-8(1)  
                        • NIST 800-53 Rev 5 SC-8(2)  
                        • NIST 800-53 Rev 5 SI-7(6)  
                        • Limit network access  
                        • US East (N. Virginia) arn:aws:controltower:us-east-1::control/BORGQLJMFKKU  
                        • US East (Ohio) arn:aws:controltower:us-east-2::control/LJKJKRDIYJNY  
                        • US West (Oregon) arn:aws:controltower:us-west-2::control/FEXUWRZIRBJR  
                        • Canada (Central) arn:aws:controltower:ca-central-1::control/GFVYJMUPS6XT  
                        • Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/LBDQCQLATJBB  
                        • Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/WHZUTCZDEVXU  
                        • Europe (Frankfurt) arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• SH.ELB.9</td>
<td></td>
<td></td>
<td>central-1::control/YQLCUFAOJXUX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/GADJVJLUSKE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/DSKPMWLPQDG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/IJXHXMULMKRD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/BBKARSAMZQQX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/LSDFYBAGXZRB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/DPJVNBVQG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/YYPWRLINUDDZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/VYFJFUWVFUY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/VYYVUCUSYMJD</td>
</tr>
</tbody>
</table>

**SH.ELB.9**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ELB.9</td>
<td>• NIST 800-53 Rev 5 CP-10</td>
<td>• Improve availability</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td></td>
<td>east-1::control/LYMRJYHLDSO</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-36</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/JHHGKBILPUKI</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/DFCKOHVLNFD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/NIQNQIVTOBLT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/PYRVSCBYFIYM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/UPDNDZTPZA In</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ WPWBIGNPRVNS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/VBDNCNX00KSD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/GZBBCHZOGDLP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/SDBXIGMFJFDR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/SVECMBMGSKGW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/IFCBLNRPNRNM</td>
</tr>
</tbody>
</table>
| | | | Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ELBv2.1</td>
<td>NIST 800-53 Rev 5 AC-17(2)</td>
<td>Encrypt data in transit</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/PPNNYFOYSBQX</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/LNITYBGXKRZ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 IA-5(1)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/VBIBPIJKJEYJ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-12(3)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/YGASWQGDRBGT</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/FJVEBRUCXPRO</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-23</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AJRCPZGMLTZJ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-23(3)</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td>d-53 Rev 5 SC-8(1)</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.EMR.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/BSHPLTXRABUV</td>
</tr>
</tbody>
</table>

- PCI DSS version 3.2.1 2.3
- PCI DSS version 3.2.1 4.1
- PCI DSS version 3.2.1 8.2.1
- Europe (Ireland) arn:aws:controltower:eu-west-1::control/PKGSEAY0WAOJ
- Europe (London) arn:aws:controltower:eu-west-2::control/LOWUMPQRPIGR
- Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ZQCDKAMUINPZ
- Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/BATAGMQWYONM
- Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/KCWNEMFFUYLN
- Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/JNZLKQTNNUFG
- Europe (Paris) arn:aws:controltower:eu-west-3::control/GWLFGQOFTRSI
- South America (São Paulo) arn:aws:controltower:sa-east-1::control/BSHPLTXRABUV
- US West (N. California) arn:aws:controltower:us-west-1::control/JDTAAUX3TEYF

SH.EMR.1

- NIST 800-53 Rev 5 AC-21
- Limit network access

arn:aws:controltower:us-east-1::control/BSHPLTXRABUV
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td></td>
<td>east-1::control/IFSMMPPPVKOTA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• US East (Ohio) arn:aws:controltower:us-east-2::control/JETOPUCSYFAI
• US West (Oregon) arn:aws:controltower:us-west-2::control/JGCWFSRMUIIO
• Canada (Central) arn:aws:controltower:ca-central-1::control/EPAGONAIKWZGV
• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/G3CORWHPFGQY
• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/LCGUHRCWQNI
• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/RBAJHFBQNIQ
• Europe (Ireland) arn:aws:controltower:eu-west-1::control/GVNSLDLJUOON
• Europe (London) arn:aws:controltower:eu-west-2::control/IGGIYASTPRJG
• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/IBLZUDBNMMLT
• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/NOTXPWZXUBGQ
• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/FUWGXDEJHVL
• Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ES.1</td>
<td></td>
<td>Encrypt data at rest</td>
<td>northeast-1::control/SBGVRVKPCNKE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/SEFFPSXRZCDD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(São Paulo) arn:aws:controltower:sa-east-1::control/IMKKFPVGPVRHC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/FKPHOKJPJXQV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ES.1</td>
<td></td>
<td>Encrypt data at rest</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/NKJEYRKBVFV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/UCHAADGIVPUY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/MDTDTZWSIGIP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/POZXDXUYCPRU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/SFSPFDZNCYNL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Singapore) arn:aws:controltower:ap-southeast-1::control/BIMQVMQISKP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eucentral-1::control/SBECRVRVKPCNKE</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>SH.ES.2</td>
<td></td>
<td></td>
<td>central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PFHFMPYLFGRU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FMHZDKQFRJTA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GQVXDQQIVYOG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PTTSHGGLKLPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SCVOJNYHGDOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LASVWDCILYGQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>UMFDEWAMMCFN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>KZBVIPYLQACH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>QHWIQGMZHJS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ICJDPHYDJMB</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td></td>
<td>east-1::control/QCHIJWURAOFY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US East (Ohio)</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ZOTKTSFENTUJ</td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/HDSBLYRMYRX</td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:central-1::control/SZZLDHEGCRE</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/BAQAAMCLMWUE</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/FUGYRQZUEJYV</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:europe-central-1::control/BOLPWJIEXNK</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:europe-west-1::control/EIAQGIKVMUXC</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:europe-west-2::control/JAQAMUDGAIVW</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:europe-north-1::control/LRNCPQEFSSY</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/NOXXLSV5ZDSE</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/TPVQKJ0DFVL</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo) | | | Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.ES.3            | • NIST 800-53 Rev 5 AC-4  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-23  
• NIST 800-53 Rev 5 SC-23(3)  
• NIST 800-53 Rev 5 SC-7(4)  
• NIST 800-53 Rev 5 SC-8  
• NIST 800-53 Rev 5 SC-8(1)  
• NIST 800-53 Rev 5 SC-8(2)  
• PCI DSS version 3.2.1 4.1 | • Encrypt data in transit | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/  
QYNCJNDFTBXE  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/  
ULPXARUKLUVO  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/  
FMBZHHQPTZTW  
• Canada (Central)  
ar:n:aws:controltower:ca-central-1::control/  
XLMZTTAZIZFJ  
• Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/  
LDLVCU0J3CXV  
• Asia Pacific (Singapore)  
ar:n:aws:controltower:ap-southeast-1::control/  
PPYOPHNGFCAW  
• Europe (Frankfurt)  
ar:n:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ES.4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SH.ES.4

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-4(26)</td>
<td></td>
<td></td>
<td>east-1::control/STVVFHXXKVQM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(9)</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/BWPW0ZTDEDRN</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/SJLJVDZSATIT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/DHRRWFEW8KNT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/NTWHSQI0KOPQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/GFKAIZZPILLQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/LTWJGEJEMYGH</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/RMAQOLXGSLKK</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/RVIBRAMTLWXY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/WZYPFCRTWXIX</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/TIPBMSRPM0NC</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/SYUVAMHWZNFNM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-</td>
</tr>
</tbody>
</table>
### Control API identifiers, by Region

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.ES.5            | • PCI DSS version 3.2.1 10.3.5  
• PCI DSS version 3.2.1 10.3.6 | Establish logging and monitoring | nordest-1::control/HNFCSNRZEIRX  
• Europe (Paris)  
ar:naws:controltower:eu-west-3::control/NCUUHDFHMXWJ  
• South America (São Paulo)  
ar:naws:controltower:saeast-1::control/HHDZXJRNTOKS  
• US West (N. California)  
ar:naws:controltower:uswest-1::control/PWMKwZWORUTC |

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.ES.5            | • NIST 800-53 Rev 5 AC-2(4)  
• NIST 800-53 Rev 5 AC-4(26)  
• NIST 800-53 Rev 5 AC-6(9)  
• NIST 800-53 Rev 5 AU-10  
• NIST 800-53 Rev 5 AU-12  
• NIST 800-53 Rev 5 AU-2  
• NIST 800-53 Rev 5 AU-3  
• NIST 800-53 Rev 5 AU-6(3)  
• NIST 800-53 Rev 5 AU-6(4)  
• NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 SC-7(9)  
• NIST 800-53 Rev 5 SI-3(8) | Establish logging and monitoring | • US East (N. Virginia)  
ar:naws:controltower:useast-1::control/MMCPMMUFJYU  
• US East (Ohio)  
ar:naws:controltower:useast-2::control/XMJYMFUZYCDL  
• US West (Oregon)  
ar:naws:controltower:uswest-2::control/CJEQHRVXTIWI  
• Canada (Central)  
ar:naws:controltower:ca-central-1::control/FUEFZWKCJABK  
• Asia Pacific (Sydney)  
ar:naws:controltower:ap-southeast-2::control/HBJICLMPQAQ0  
• Asia Pacific (Singapore)  
ar:naws:controltower:ap-southeast-1::control/AKW5XUIZDGPZ  
• Europe (Frankfurt)  
ar:naws:controltower:europe-central-1::control/JHCWZPAWZQ |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ES.6</td>
<td>• NIST 800-53 Rev 5</td>
<td>• Improve availability</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td>SI-4(20)</td>
<td></td>
<td>east-1::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SI-7(8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.2.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.2.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.2.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.2.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.3.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.3.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.3.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SI-10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SH.ES.6**
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td></td>
<td>east-1::control/ RVPSMUHBMQH</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-36</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ URI00ADAQOQPW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ WKCJFNZAPCDY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ FJDN5WEYJTV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ LSVJLTGRQG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ GSKVJECFCYUL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-36</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ THCGKVUQLBQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ IJBRHMLEUQOW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ GZYPQIA586</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-36</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ GLOMYATYELZ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ KSBFZAUUIG</td>
</tr>
</tbody>
</table>
| • NIST 800-53 Rev 5 SC-5(2) | | | Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.E.S.7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **NIST 800-53 Rev 5 CP-10**  
- **NIST 800-53 Rev 5 CP-6(2)**  
- **NIST 800-53 Rev 5 SC-36**  
- **NIST 800-53 Rev 5 SC-5(2)**  
- **NIST 800-53 Rev 5 SI-13(5)**  
- Improve availability  
- **US East (N. Virginia)**  
  - arn:aws:controltower:us-east-1::control/BHBTVOFTLC5B  
- **US East (Ohio)**  
  - arn:aws:controltower:us-east-2::control/KAFELHVBFCBK  
- **US West (Oregon)**  
  - arn:aws:controltower:us-west-2::control/KDDVJHAFANAS  
- **Canada (Central)**  
  - arn:aws:controltower:ca-central-1::control/RHTIIINVLDRG  
- **Asia Pacific (Sydney)**  
  - arn:aws:controltower:ap-southeast-2::control/YKZLIAIKUETJ  
- **Asia Pacific (Singapore)**  
  - arn:aws:controltower:ap-southeast-1::control/QORXBYK0QWGOS  
- **Europe (Frankfurt)**  
  - arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ES.8</td>
<td></td>
<td>Encrypt data in transit</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-west-1::control/ QHEQAAHVJ3SP</td>
</tr>
</tbody>
</table>

**SH.ES.8**

- **Control identifier**: SH.ES.8
- **Framework**: NIST 800-53 Rev 5 AC-17(2)
- **Control objective**: Encrypt data in transit
- **Control API identifiers, by Region**: arn:aws:controltower:us-east-1::control/ QHEQAAHVJ3SP
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 IA-5(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-12(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-23(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- east-1::control/FZIUUKFDZJJ0
- US East (Ohio)
  arn:aws:controltower:us-east-2::control/AUHGLFUEHAPR
- US West (Oregon)
  arn:aws:controltower:us-west-2::control/QVHFDLZIKRZW
- Canada (Central)
  arn:aws:controltower:ca-central-1::control/EJOTPAVNDIFJH
- Asia Pacific (Sydney)
  arn:aws:controltower:ap-southeast-2::control/CHITVZCWFAK
- Asia Pacific (Singapore)
  arn:aws:controltower:ap-southeast-1::control/VWZXXJAVDVI
- Europe (Frankfurt)
  arn:aws:controltower:eu-central-1::control/OPQGRNKSJRBMF
- Europe (Ireland)
  arn:aws:controltower:eu-west-1::control/KLGGCWMZGVI
- Europe (London)
  arn:aws:controltower:eu-west-2::control/ISVOJHDBVFJJ
- Europe (Stockholm)
  arn:aws:controltower:eu-north-1::control/SIBUGFYSXGEW
- Asia Pacific (Mumbai)
  arn:aws:controltower:ap-south-1::control/QVPGWCRCBGPS
- Asia Pacific (Seoul)
  arn:aws:controltower:ap-northeast-2::control/WZWUGROTVPXC
- Asia Pacific (Tokyo)
  arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.ElasticBeanstalk.1 | • NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 SI-2 | • Improve resiliency | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
HFVMYSGPEYDN  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
VVCFXNZNIDRP  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
CUVOLGXTISMS  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
BDAQDLACRNUJ  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
IVNFFJMNHDXC  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
QTKGH0CLSKQR  
• Europe (Frankfurt)  
arn:aws:controltower:eu- |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.ElasticBeanstalk.2</td>
<td>NIST 800-53 Rev 5 SI-2</td>
<td>Manage vulnerabilities</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ZBUJLIWPXMZC</td>
</tr>
</tbody>
</table>

**SH.ElasticBeanstalk.2**
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(2)</td>
<td></td>
<td></td>
<td>east-1::control/CIYPAZNJKRRG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(4)</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/UPAWTQLRRSIX</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(5)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/IJZKGIDVPZUE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 6.2</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/WIFQEAWBYYY</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ALIABBGIBJQX</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/UZMJMWJXYBRE</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/WIZMLNWMLPL</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/OTKEBOVKUWDW</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/LWIJCVNZKWGE</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/HWWRWNQMayGv</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/DWGNUMIILACZO</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/OLBWUHFKLXTO</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.GuardDuty.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-2(12)</td>
<td>• Prepare for incident response</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/HHVKNVANXXAA</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-6(1)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/LDUGVICWFOGH</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-6(5)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/APFIGXPGUEUMB</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/MPEMCZUTKJSC</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-8(3)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/SEPYALESYFHN</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 RA-3(4)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/NOLGMPH0APVF</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SA-11(1)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-</td>
</tr>
</tbody>
</table>
### SH.IAM.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.IAM.1</td>
<td>CIS AWS Benchmark 1.4 1.16</td>
<td>Enforce least privilege</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/ ZQJRRNCGOEVOG</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-2</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ AMOSAUZUOENC</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ BYCPKIWGCXUS</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ UIZQGWRMOAM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ LFXQGUZ1JIVJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ JUJQVSCZAMM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-5</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ RWLDODTORAP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ GPMRUNIPAAHZ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(10)</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ XZFJHDJ2ZVRN</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(2)</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ AFVWEIUEO</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(3)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ TENOMRDYBHFD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ MFXIIBSZNHHF</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------</td>
<td>----------------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.IAM.2           | • CIS AWS Benchmark 1.4 1.15  
                      • NIST 800-53 Rev 5 AC-2  
                      • NIST 800-53 Rev 5 AC-2(1)  
                      • NIST 800-53 Rev 5 AC-3  
                      • NIST 800-53 Rev 5 AC-3(15)  
                      • NIST 800-53 Rev 5 AC-3(7)  
                      • NIST 800-53 Rev 5 AC-5  
                      • NIST 800-53 Rev 5 AC-6  
                      • NIST 800-53 Rev 5 AC-6(10)  
                      • NIST 800-53 Rev 5 AC-6(2)  
                      • NIST 800-53 Rev 5 AC-6(3)  
                      • PCI DSS version 3.2.1 2.2 | • Enforce least privilege | northeast-1::control/WEFQVTUHENVP
• Europe (Paris)  
  arn:aws:controltower:eu-west-3::control/CFRGOUIMJJDIN
• South America  
  (São Paulo)  
  arn:aws:controltower:south-america-east-1::control/RDFMPAZPDBSW
• US West (N. California)  
  arn:aws:controltower:us-west-1::control/PPYXQOKDTPWB
• US East (N. Virginia)  
  arn:aws:controltower:us-east-1::control/CKRCAIBXILFK
• US East (Ohio)  
  arn:aws:controltower:us-east-2::control/GNUNOTQMCLNP
• US West (Oregon)  
  arn:aws:controltower:us-west-2::control/VBOICSFJFYQV
• Canada (Central)  
  arn:aws:controltower:ca-central-1::control/BQESNVPCOBGD
• Asia Pacific (Sydney)  
  arn:aws:controltower:asia-pacific-southeast-2::control/YKNIXGERUKGE
• Asia Pacific  
  (Singapore)  
  arn:aws:controltower:asia-pacific-southeast-1::control/NVWWQVOKPQZU
• Europe (Frankfurt)  
  arn:aws:controltower:eu-
## SH.IAM.21

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.IAM.21          | NIST 800-53 Rev 5 AC-2   | Enforce least privilege      | US East (N. Virginia)
|                    |                          |                              |  arn:aws:controltower:us-east-1::control/   |
|                    |                          |                              | Q0UCRVYJWUO                                |

### SH.IAM.21 Details:

- **Control identifier**: SH.IAM.21
- **Framework**: NIST 800-53 Rev 5 AC-2
- **Control objective**: Enforce least privilege
- **Control API identifiers, by Region**:
  - US East (N. Virginia):
    - arn:aws:controltower:us-east-1::control/Q0UCRVYJWUO
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td></td>
<td></td>
<td>east-1::control/HQVLATIMVODA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/ WUAKZMFMASZA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ FEPDQJBEHJE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-7</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ PPLFAEVCJSZX</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-5</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ CWYTGQWDYNL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ WUHBCQNMVRD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(10)</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ VIPDQGLETFRV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(2)</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/ PQTGHIEPD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(3)</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ USZNDBRQZJR</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ HCJUBNEAEYCY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ WDVNBMRNREIU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.2</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ MXPBNMAHMHQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.IAM.3 | • CIS AWS Benchmark 1.4 1.14  
• NIST 800-53 Rev 5 AC-2(1)  
• NIST 800-53 Rev 5 AC-2(3)  
• NIST 800-53 Rev 5 AC-3(15)  
• PCI DSS version 3.2.1 2.2  
• PCI DSS version 3.2.1 8.2.4 | • Enforce least privilege | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/ZEJBJAESPYR |  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/DWHOWHNGILYZ |  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/NJX5DDNVGFJT |  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/ZBRPUSKMGALZ |  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/ZWZXSXOWDRJN |  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/ZQDCZNSVLUWW |  
• Europe (Frankfurt)  
arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.IAM.4</td>
<td></td>
<td></td>
<td>central-1::control/ NYwZTJUICRUT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ RHVPZJVBRVIC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ WBMPXAAQFWFJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ QCUyUTVQDRTV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ RFHKMYZYMHY0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ MCTNeB TxCmpf</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ EGBxSeL3JNDTO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/ XQFZKZXQPJLG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/ TINYBAUSDLa</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/ DGSPIXKNFkxW</td>
</tr>
</tbody>
</table>

**SH.IAM.4**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.IAM.4</td>
<td>• CIS AWS Benchmark 1.4 1.4</td>
<td>• Enforce least privilege</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.IAM.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWS Control Tower User Guide</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **NIST 800-53 Rev 5**
  - AC-2(1)
  - AC-3(15)
  - AC-3(7)
  - AC-6
  - AC-6(10)
  - AC-6(2)
  - PCI DSS version 3.2.1
    - 2.2
    - 2.1
    - 7.1.1
    - 7.1.2
    - 7.2.1
    - 7.2.2
    - 8.1.1
  - west-2:
    - control/MLZMKASTNAMJ
    - control/BWMCZWY0POFV
    - control/XMQHVQYMQVANU
    - control/LFHALIRSAOMG
    - control/CVTQQOH8XJYE
    - control/KRKFUZIAZCH
    - control/FVFAHHXHVEDZ
    - control/NEALLBIMAHLJ
    - control/PQZZNTWBDMA
    - control/WMMELOBHWP
    - control/MYNFRQFMIDRC
    - control/KNGXYOAJBAD
    - control/SH.IAM.4
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.IAM.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Control identifier**

- SH.IAM.5

**Framework**

- CIS AWS Benchmark 1.4 1.10
- NIST 800-53 Rev 5 AC-2(1)
- NIST 800-53 Rev 5 AC-3(15)
- NIST 800-53 Rev 5 IA-2(1)
- NIST 800-53 Rev 5 IA-2(2)
- NIST 800-53 Rev 5 IA-2(6)
- NIST 800-53 Rev 5 IA-2(8)
- PCI DSS version 3.2.1 2.2
- PCI DSS version 3.2.1 8.3.1
- PCI DSS version 3.2.1 8.3.2
- PCI DSS version 3.2.1 8.6

**Control objective**

- Enforce least privilege

**Control API identifiers, by Region**

- **northeast-1::control/LCWHRMFYRDPY**
- **Europe (Paris)** arn:aws:controltower:eu-west-3::control/JQIXBFLCEXKN
- **South America (São Paulo)** arn:aws:controltower:saeast-1::control/DYTPSGETQHGZ
- **US West (N. California)** arn:aws:controltower:us-west-1::control/KRXFGDZBYIR
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.IAM.6</td>
<td></td>
<td></td>
<td>Central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JZQVFKSOFWSA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WUGDSPITURNA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>YWKENJULZUI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HXUNMULYFTBQJC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MAJHPOEPBKQN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ECYNNUZFPWMI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GCNKIMWWZRKL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WXQUFEUXQQLB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RKTXSSCRCSOG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CWPEYHMAQLIX</td>
</tr>
</tbody>
</table>

**SH.IAM.6**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.IAM.6</td>
<td>CIS AWS Benchmark 1.4 1.6</td>
<td>• Enforce least privilege</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td></td>
<td></td>
<td>east-1::control/YXQAYRGZITSE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/VMFIYFNTTOHK</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 IA-2(1)</td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/MEFQIQZLYFAO</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 IA-2(2)</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/RBIDFUAIDOZW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 IA-2(6)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/DJMVZWKQL0WP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 IA-2(8)</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/PUNPGEQKLGNW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/NKLWDWNZYKALI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.3.1</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/FSWHMTMWCPCU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.3.2</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/EQQYPBMUXUZP</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.3.2</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/RIYGDPNPHQHD</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/HIVWGOODPEDII</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/BOIRCXQEYNLQ</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo) | | | Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.IAM.7           | • CIS AWS Benchmark 1.4 1.8  
• CIS AWS Benchmark 1.4 1.9  
• NIST 800-53 Rev 5 AC-2(1)  
• NIST 800-53 Rev 5 AC-2(3)  
• NIST 800-53 Rev 5 AC-3(15)  
• NIST 800-53 Rev 5 IA-5(1)  
• PCI DSS version 3.2.1 8.2.3  
• PCI DSS version 3.2.1 8.2.4  
• PCI DSS version 3.2.1 8.2.5  | • Protect configurations | northeast-1::control/FFFZNTDNXQVY  
- Europe (Paris)  
  arn:aws:controltower:eu-west-3::control/WTTOHMYBKCFD  
- South America (São Paulo)  
  arn:aws:controltower:saeast-1::control/ZDWYGZQRVGUZ  
- US West (N. California)  
  arn:aws:controltower:us-west-1::control/XPMKIIIBTCPX |
|                    | • US East (N. Virginia)  
  arn:aws:controltower:useast-1::control/YJDQWPMKBQWV  
• US East (Ohio)  
  arn:aws:controltower:useast-2::control/QJKCDNAHSGNU  
• US West (Oregon)  
  arn:aws:controltower:us-west-2::control/VBKLNMETHZB  
• Canada (Central)  
  arn:aws:controltower:ca-central-1::control/ZUYARIGWZBSL  
• Asia Pacific (Sydney)  
  arn:aws:controltower:ap-southeast-2::control/XLUHJBTPSEGR  
• Asia Pacific (Singapore)  
  arn:aws:controltower:ap-southeast-1::control/GLWFPUPESYOC  
• Europe (Frankfurt)  
  arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.IAM.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IQPAHZJVRRMC</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>Europe (Ireland)</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>Europe (London)</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>Europe (Stockholm)</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul)</td>
</tr>
<tr>
<td>• Asia Pacific</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>Europe (Paris)</td>
</tr>
<tr>
<td>• South America</td>
<td></td>
<td></td>
<td>South America (São Paulo)</td>
</tr>
<tr>
<td>• US West (N.</td>
<td></td>
<td></td>
<td>US West (N. California)</td>
</tr>
<tr>
<td>California)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SH.IAM.8

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.IAM.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• CIS AWS Benchmark</td>
<td>1.4 1.12</td>
<td>• Enforce least</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>privilege</td>
<td>arn:aws:controltower:us-west-1::control/GJMEDGOCQHMC</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-2</td>
<td></td>
<td></td>
<td>east-1::control/DYZBFPIWFDCL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/JQZWJZWMTQCQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-2(3)</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/VUQQVIQVRPBS</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/SDKZNRQZVJYU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/HEXEUELTWUJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ALXCCRNIHSHQ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/XZDDELXKYDNY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/OTDQBWDVYVJE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.1.4</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/UNYQJZPHYTQB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ABCVAFYFOUZL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/FKPJQKAODREW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/LLXPAYEHWBTA</td>
</tr>
</tbody>
</table>
|  |  |  | • Asia Pacific (Tokyo) arn:aws:controltower:ap-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="https://www.nist.gov/800-53">NIST 800-53 Rev 5 AC-5</a></td>
<td></td>
<td>• Asia Pacific (Singapore) <a href="https://aws.amazon.com/">arn:aws:controltower:ap-southeast-1::control/</a> IWKWHAGMHFQU</td>
</tr>
<tr>
<td></td>
<td><a href="https://www.nist.gov/800-53">NIST 800-53 Rev 5 AC-6</a></td>
<td></td>
<td>• Europe (Frankfurt) [arn:aws:controltower:eu-]</td>
</tr>
<tr>
<td></td>
<td><a href="https://www.pcisecuritystandards.org">PCI DSS version 3.2.1 3.5.2</a></td>
<td></td>
<td>• US East (Ohio) <a href="https://aws.amazon.com/">arn:aws:controltower:us-east-2::control/</a> NUJRZCEGSRSQ</td>
</tr>
<tr>
<td></td>
<td><a href="https://www.pcisecuritystandards.org">PCI DSS version 3.2.1 7.1.1</a></td>
<td></td>
<td>• US West (Oregon) <a href="https://aws.amazon.com/">arn:aws:controltower:us-west-2::control/</a> OONPBHUUNDTP</td>
</tr>
<tr>
<td></td>
<td><a href="https://www.pcisecuritystandards.org">PCI DSS version 3.2.1 7.1.2</a></td>
<td></td>
<td>• Canada (Central) <a href="https://aws.amazon.com/">arn:aws:controltower:ca-central-1::control/</a> VZYJNYATGYQB</td>
</tr>
<tr>
<td></td>
<td><a href="https://www.pcisecuritystandards.org">PCI DSS version 3.2.1 7.2.1</a></td>
<td></td>
<td>• Asia Pacific (Sydney) <a href="https://aws.amazon.com/">arn:aws:controltower:ap-southeast-2::control/</a> YBJERSBVCEHS</td>
</tr>
<tr>
<td></td>
<td><a href="https://www.pcisecuritystandards.org">PCI DSS version 3.2.1 7.2.1</a></td>
<td></td>
<td>• Asia Pacific (Singapore) <a href="https://aws.amazon.com/">arn:aws:controltower:ap-southeast-1::control/</a> IWKWHAGMHFQU</td>
</tr>
<tr>
<td></td>
<td><a href="https://www.pcisecuritystandards.org">PCI DSS version 3.2.1 7.2.1</a></td>
<td></td>
<td>• Europe (Frankfurt) [arn:aws:controltower:eu-]</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>SH.KMS.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td></td>
<td>central-1::control/QGQGZMRWIRVZ</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/YIKIWDHYLGPR</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/ZHHPIYVVGYP</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/QKTHXYQPLNYY</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/IGBZSWRYBKOJ</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/HEQYHQMETFWE</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/BYXFJJHFUZOU</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/GTNCXJXJAMYN</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:saeast-1::control/IIMAA0EMTFUC</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/BWCBIWNWVGXU</td>
</tr>
</tbody>
</table>

**SH.KMS.2**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.KMS.2</td>
<td>• NIST 800-53 Rev 5 AC-2</td>
<td>• Enforce least privilege</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-west-1::control/</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td></td>
<td></td>
<td>east-1::control/QNVYEARPCKHY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/LTNDXSCGBACC</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/KOWUMYVGIFEUY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-7(7)</td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/IXRRCZUELMDU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/NHKYZBGUEQPPJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AC-6(3)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/KSMDOJXHBPMW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.5.2</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/YGKEXZMFPRPN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/KPKAHTFYDJJQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.2</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/PPKCO1ZMWYXC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/WTDVCVDHUVVI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/JBH0ZLAAZRRH</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.3</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/CUVJHPIMJUHF</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.4</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td><strong>SH.KMS.3</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Control identifier</strong></td>
<td><strong>Framework</strong></td>
<td><strong>Control objective</strong></td>
<td><strong>Control API identifiers, by Region</strong></td>
</tr>
</tbody>
</table>
| SH.KMS.3            | • NIST 800-53 Rev 5 SC-12  
|                     | • NIST 800-53 Rev 5 SC-12(2)  
|                     | • PCI DSS version 3.2.1 3.6.5  
|                     | • PCI DSS version 3.2.1 3.6.7 | • Protect data integrity | • US East (N. Virginia)  
|                     |                     |                              | arn:aws:controltower:us-east-1::control/CKCEVZPVWRJV |
|                     |                     |                              | • US East (Ohio)  
|                     |                     |                              | arn:aws:controltower:us-east-2::control/HKNGPDPGWTEN |
|                     |                     |                              | • US West (Oregon)  
|                     |                     |                              | arn:aws:controltower:us-west-2::control/JNOLPJAAZDRU |
|                     |                     |                              | • Canada (Central)  
|                     |                     |                              | arn:aws:controltower:ca-central-1::control/QRKASNXFSCTZ |
|                     |                     |                              | • Asia Pacific (Sydney)  
|                     |                     |                              | arn:aws:controltower:ap-southeast-2::control/BDIQMPNVEKNT |
|                     |                     |                              | • Asia Pacific (Singapore)  
|                     |                     |                              | arn:aws:controltower:ap-southeast-1::control/TUHCPYQGZEHJ |
|                     |                     |                              | • Europe (Frankfurt)  
<p>|                     |                     |                              | arn:aws:controltower:eu- |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Kinesis.1</td>
<td></td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia) arn:aws:controltower:us-west-1::control/WVDONOWFFERT</td>
</tr>
</tbody>
</table>

**SH.Kinesis.1**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Kinesis.1</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia) arn:aws:controltower:us-west-1::control/WVDONOWFFERT</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 CM-3(6)</td>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td>east-1::control/ LKIISSYYCVINO</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-28(1)</td>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ LFMGOMWNW1ZYW</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5 SC-28</td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ XWJFEU5B2YWC</td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ MGBKMMMLYFDEG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ L3RFKNNZTFU</td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ YTWDULHCPTP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ OZCCICXZVLAG</td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ OXTKQQPJYIB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ DKOUDYFDPEJG</td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ PREPRJHMCLIQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ YPEW5ZCWJ1UR</td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ WTQVFPGFVQFNL</td>
</tr>
</tbody>
</table>
| • NIST 800-53 Rev 5 SC-28 | | • Asia Pacific (Tokyo) arn:aws:controltower:ap-
### SH.Lambda.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.Lambda.1        | • NIST 800-53 Rev 5 AC-21  
                   |           | • Limit network access | • US East (N. Virginia)  
                   |           |                               | arn:aws:controltower:us-east-1::control/  
                   |           |                               | SRLPVJHNHYTD |
|                    | • NIST 800-53 Rev 5 AC-3  
                   |           |                               | • US East (Ohio)  
                   |           |                               | arn:aws:controltower:us-east-2::control/  
                   |           |                               | ORSOXPSSFMMIB |
|                    | • NIST 800-53 Rev 5 AC-3(7)  
                   |           |                               | • US West (Oregon)  
                   |           |                               | arn:aws:controltower:us-west-2::control/  
                   |           |                               | BNIHRCGLGNHH |
|                    | • NIST 800-53 Rev 5 AC-4  
                   |           |                               | • Canada (Central)  
                   |           |                               | arn:aws:controltower:ca-central-1::control/  
                   |           |                               | BOWRFRNKVWJH |
|                    | • NIST 800-53 Rev 5 AC-4(21)  
                   |           |                               | • Asia Pacific (Sydney)  
                   |           |                               | arn:aws:controltower:ap-southeast-2::control/  
                   |           |                               | UQXXELUQDDBO |
|                    | • NIST 800-53 Rev 5 AC-6  
                   |           |                               | • Asia Pacific (Singapore)  
                   |           |                               | arn:aws:controltower:ap-southeast-1::control/  
                   |           |                               | VTLSMHLQZNAJ |
|                    | • NIST 800-53 Rev 5 SC-7  
                   |           |                               | • Europe (Frankfurt)  
                   |           |                               | arn:aws:controltower:eu- |
### SH.Lambda.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Lambda.2</td>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td>• Manage vulnerabilities</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/1234567890123</td>
</tr>
</tbody>
</table>

**AWS Control Tower User Guide**
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
<td>east-1::control/IOGKXEBRD6RUW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US East (Ohio)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/MUKLGCDYGHQH</td>
</tr>
<tr>
<td>• US West (Oregon)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/KIONODJWVSXV</td>
</tr>
<tr>
<td>• Canada (Central)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/FCQWRALM2QGM</td>
</tr>
<tr>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/TOCWOWCANSFU</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/NPXHIQYCWKOR</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/WLFORDCRAIPB</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/PIGDBBNREIU</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/ETXINVEIROP</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/SNXJDQUCDOAM</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/BHBSEUQUGFWL</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/QPNEMKAWAIIV</td>
</tr>
</tbody>
</table>
| • Asia Pacific (Tokyo) |  |  | arn:aws:controltower:ap-
## Control identifier, Framework, Control objective, Control API identifiers, by Region

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Lambda.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **NIST 800-53 Rev 5 CP-10**
- **NIST 800-53 Rev 5 CP-6(2)**
- **NIST 800-53 Rev 5 SC-36**
- **NIST 800-53 Rev 5 SC-5(2)**
- **NIST 800-53 Rev 5 SI-13(5)**

- **Improve availability**

- **US East (N. Virginia)**
  arn:aws:controltower:us-east-1::control/MHEKRUMWQLBJ

- **US East (Ohio)**
  arn:aws:controltower:us-east-2::control/LWOJRFFJEDHR

- **US West (Oregon)**
  arn:aws:controltower:us-west-2::control/QTBXGAAGFVRJ

- **Canada (Central)**
  arn:aws:controltower:ca-central-1::control/XFYQBLRUDYNB

- **Asia Pacific (Sydney)**
  arn:aws:controltower:ap-southeast-2::control/WTXQRFALXGYL

- **Asia Pacific (Singapore)**
  arn:aws:controltower:ap-southeast-1::control/BNVVZXALMUPF

- **Europe (Frankfurt)**
  arn:aws:controltower:eu-
### SH.Neptune.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Neptune.1</td>
<td></td>
<td>Encrypt data at rest</td>
<td>US East (N. Virginia) arn:aws:controltower:us-west-1::control/JSXPJEPESCWV</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td><a href="#">NIST 800-53 Rev 5 SC-13</a></td>
<td></td>
<td>east-1::control/EKIRHGBAAOK</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td><a href="#">NIST 800-53 Rev 5 SI-7(6)</a></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/JWHKRTXDDMQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.4</td>
<td><a href="#">PCI DSS version 3.2.1 8.2.1</a></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/PWBVw5SwBTQAD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td>(Asia Pacific (Sydney)) arn:aws:controltower:ap-southeast-2::control/YIRYwJNNNLLE</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/QYYYRRLWTOGM</td>
</tr>
<tr>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/OTZEVSPPCPAV</td>
<td>(Europe (Ireland) arn:aws:controltower:eu-west-1::control/GDXCMPTLMSY</td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/NWDVPEPKHXXD</td>
</tr>
<tr>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/YGCCJHPDGFFA</td>
<td>(Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/YDRDBHSTEFF</td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/HKGQLNX0UMFP</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-</td>
<td></td>
<td></td>
<td>2314</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------</td>
<td>------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.Neptune.2       | • NIST 800-53 Rev 5 AC-2(4)  
• NIST 800-53 Rev 5 AC-4(26)  
• NIST 800-53 Rev 5 AC-6(9)  
• NIST 800-53 Rev 5 AU-10  
• NIST 800-53 Rev 5 AU-12  
• NIST 800-53 Rev 5 AU-2  
• NIST 800-53 Rev 5 AU-3  
• NIST 800-53 Rev 5 AU-6(1)  
• NIST 800-53 Rev 5 AU-6(3)  
• NIST 800-53 Rev 5 AU-6(4)  
• NIST 800-53 Rev 5 AU-6(5)  
• NIST 800-53 Rev 5 AU-7(1)  | • Establish logging and monitoring | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/ NZSQQKEQDXLV  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/ QHVCYLPZDPBEK  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/ JWQLYMAHKL0  
• Canada (Central)  
ar:n:aws:controltower:can-central-1::control/ NUNSEOUQVSXL  
• Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/ FUPOVEEUSWA  
• Asia Pacific (Singapore)  
ar:n:aws:controltower:ap-southeast-1::control/ GUOLSHIUXWMQM  
• Europe (Frankfurt)  
ar:n:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>AU-9(7)</td>
<td></td>
<td>central-1::control/</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>CA-7</td>
<td></td>
<td>ZPQKOVBBQWRP</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SC-7(9)</td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/NEEDBPYKGCN</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SI-3(8)</td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/UMZNLNFREFXWU</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SI-4(20)</td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ZPZKOGHBUPR</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SI-4(5)</td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/HQGZCYUDHVRM</td>
</tr>
<tr>
<td>NIST 800-53 Rev 5</td>
<td>SI-7(8)</td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/SKHXOXXAWZZH</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>10.1</td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/IUVQALLVXGZW</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>10.2.1</td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-east-1::control/PJBDTIYTINLGL</td>
</tr>
<tr>
<td>PCI DSS version 3.2.1</td>
<td>10.2.2</td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/BUNWDCJNVGGI</td>
</tr>
</tbody>
</table>

2316
## SH.Neptune.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.Neptune.3       | • NIST 800-53 Rev 5 AC-21  
                      • NIST 800-53 Rev 5 AC-3  
                      • NIST 800-53 Rev 5 AC-3(7)  
                      • NIST 800-53 Rev 5 AC-4  
                      • NIST 800-53 Rev 5 AC-4(21)  
                      • NIST 800-53 Rev 5 AC-6  
                      • NIST 800-53 Rev 5 SC-7  
                      • NIST 800-53 Rev 5 SC-7(11)  
                      • NIST 800-53 Rev 5 SC-7(16)  
                      • NIST 800-53 Rev 5 SC-7(20)  
                      • NIST 800-53 Rev 5 SC-7(21)  
                      • NIST 800-53 Rev 5 SC-7(3)  
                      • NIST 800-53 Rev 5 SC-7(4)  
                      • NIST 800-53 Rev 5 SC-7(9)  
                      • PCI DSS version 3.2.1 1.2.1  
                      • PCI DSS version 3.2.1 1.3  
                      • PCI DSS version 3.2.1 1.3.1  
                      • PCI DSS version 3.2.1 1.3.2  
                      • PCI DSS version 3.2.1 1.3.4  
                      • PCI DSS version 3.2.1 1.3.6  
                      • PCI DSS version 3.2.1 2.2.2  | • Enforce least privilege | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
SBCVFDUYGKYV  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
KJDLRVEAIVSYB  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
QXWJVOYQNEKQ  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
UPGOTKZMWQPV  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
GPRJFLJKTBB  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
QTLTUBCUAGTR  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/  
ZKDTBATDAALE  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/  
GPMZXXK0J0EE  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/  
YXJIW5XJEKIM  
• Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/  
EREEUEZTQVI  
• Asia Pacific (Mumbai)  
arn:aws:controltower:ap-south-1::control/  
MWNOLNFGIQDK |
### SH.Neptune.4

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.Neptune.4       |           | • Improve availability  
                     |                     | • Protect configurations       |
|                    |           | • NIST 800-53 Rev 5  
                     |                     | • US East (N. Virginia)         |
|                    |           | CA-9(1)           | • NIST 800-53 Rev 5               |
|                    |           | CM-2              | • NIST 800-53 Rev 5               |
|                    |           | CM-2(2)           | • NIST 800-53 Rev 5               |
|                    |           | CM-3              | • NIST 800-53 Rev 5               |
|                    |           | SC-5(2)           | • NIST 800-53 Rev 5               |
|                    |           | PCI DSS version 3.2.1 2.2  | • US East (Ohio)                  |
|                    |           |                   | • US West (Oregon)                |
|                    |           |                   | • Canada (Central)                |
|                    |           |                   | • Asia Pacific (Sydney)           |
|                    |           |                   | • US East (N. Virginia)           |
|                    |           |                   | • US East (Ohio)                  |
|                    |           |                   | • US West (Oregon)                |
|                    |           |                   | • Canada (Central)                |
|                    |           |                   | • Asia Pacific (Sydney)           |

- Asia Pacific (Seoul)  
  arn:aws:controltower:ap-northeast-2::control/AGQDWLBSVZHW
- Asia Pacific (Tokyo)  
  arn:aws:controltower:ap-northeast-1::control/FTUDPPTFLVO
- Europe (Paris)  
  arn:aws:controltower:eu-west-3::control/AZVYVLXSEMIH
- South America (São Paulo)  
  arn:aws:controltower:south-america-east-1::control/JNYGVG5VCVBI
- US West (N. California)  
  arn:aws:controltower:us-west-1::control/XLFTYXPVMGTS
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/ODXEWSFKGMDG</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/KQNOJHSPSWZH</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/DMWNICKIQVGC</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/RINTJPAJKWU</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/ADORVDROWVPA</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/WYUUHCKLJMJJN</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/GYPMVKORWSCM</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/DWOABBMIWODR</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/AKZDXQMJAOAM</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/OAKVBKHIMKTC</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/RBCJUCVKLOSB</td>
</tr>
</tbody>
</table>
## SH.Neptune.5

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.Neptune.5       | • NIST 800-53 Rev 5 SI-12  
                     • PCI DSS version 3.2.1 3.1 | • Improve resiliency | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/ZYAJHMJFHKWJ  
                     • US East (Ohio) arn:aws:controltower:us-east-2::control/CKIKYVTTLNFL  
                     • US West (Oregon) arn:aws:controltower:us-west-2::control/LCPVESGNCLWT  
                     • Canada (Central) arn:aws:controltower:ca-central-1::control/DHYIFIEGKCHK  
                     • Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/IEOEHVHFUPUO  
                     • Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AHWGAETJPNSM  
                     • Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ETKRDEYMOPRO  
                     • Europe (Ireland) arn:aws:controltower:eu-west-1::control/KPAXHNZWNSWE  
                     • Europe (London) arn:aws:controltower:eu-west-2::control/DCOYCPDNYJUE  
                     • Europe (Stockholm) arn:aws:controltower:eu-north-1::control/PFZNFCAUVFCE  
                     • Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/DFLUIGICKNQB |
### SH.Neptune.6

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.Neptune.6       |           | • Encrypt data at rest | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/FUHJOTQRVJKS  
• US East (Ohio) arn:aws:controltower:us-east-2::control/OWJHHGLSLKKCZ  
• US West (Oregon) arn:aws:controltower:us-west-2::control/WGJJYXXWBSKA  
• Canada (Central) arn:aws:controltower:ca-central-1::control/VRMZNTNTYOFHJ  
• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/KMNEEERHO2WO |
|                    |           | • PCI DSS version 3.2.1 3.4  
• PCI DSS version 3.2.1 8.2.1 | • Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/VZANDNGXQTNF  
• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/OHKXCRIOQIZFT  
• Europe (Paris) arn:aws:controltower:eu-west-3::control/IPXRSSRRYRJC  
• South America (São Paulo) arn:aws:controltower:sa-east-1::control/FFMGXSRRYRJC  
• US West (N. California) arn:aws:controltower:us-west-1::control/PCCBHGJUDJZ |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/ KYGVJYXTEIYZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/ ROVIKNQOEUEA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/ FSEXTDBFHGZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ MZGGVMWREOLW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ CQHZFMPTVOVN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ RGTMIUHBCJER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ DWWVOIHDHQNZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ ZIAVWMESTETC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ MDUGRJDVHOAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ MIDTNRJVELZH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ AIGZVIHZIKTP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.Neptune.7       | • NIST 800-53 Rev 5 AC-2(1)  
• NIST 800-53 Rev 5 AC-3  
• NIST 800-53 Rev 5 AC-3(15)  
• NIST 800-53 Rev 5 AC-3(7)  
• NIST 800-53 Rev 5 AC-6  
• PCI DSS version 3.2.1 7.1.1  
• PCI DSS version 3.2.1 7.2.1  
• PCI DSS version 3.2.1 8.7 | • Use strong authentication | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/QUQXKRSLQMMU  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/CWAKJSEKLKFQ  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/WKEGTXZTMJQQ  
• Canada (Central)  
ar:n:aws:controltower:ca-central-1::control/IXZCJBXMBCGV  
• Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/ELPFKNRXK0FT  
• Asia Pacific (Singapore)  
ar:n:aws:controltower:ap-southeast-1::control/HTMETSUNKKNC  
• Europe (Frankfurt)  
ar:n:aws:controltower:eu-central-1::control/ZWMMWACAAKX  
• Europe (Ireland)  
ar:n:aws:controltower:eu-west-1::control/JUDPXQZMWMHP  
• Europe (London)  
ar:n:aws:controltower:eu-west-2::control/ ZSKMJGSZRJ0L  
• Europe (Stockholm)  
ar:n:aws:controltower:eu-north-1::control/WJIXMOGTB5R  
• Asia Pacific (Mumbai)  
ar:n:aws:controltower:ap-south-1::control/ZREIJBXJRJCO |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.Neptune.8       |           | Protect configurations | US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/VVPIJUDOURCA  
US East (Ohio)  
arn:aws:controltower:us-east-2::control/RFZSTHMECLZN  
US West (Oregon)  
arn:aws:controltower:us-west-2::control/HZVBZDZLPYJP  
Canada (Central)  
arn:aws:controltower:ca-central-1::control/MNSOQMBYSSN  
Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/FLKRVTPMLUMN |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>Arn:aws:controltower:ap-southeast-1::control/POJMDJSLIUJF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>Arn:aws:controltower:eu-central-1::control/PNMLKBCTHPLN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>Arn:aws:controltower:eu-west-1::control/IPODQTVQNOEC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td>Arn:aws:controltower:eu-west-2::control/KYNITBYARJGK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>Arn:aws:controltower:eu-north-1::control/BEUAZKYAMIKM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>Arn:aws:controltower:ap-south-1::control/IQNCAGYRQMAC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>Arn:aws:controltower:ap-northeast-2::control/WHZKUALDWLXH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>Arn:aws:controltower:ap-northeast-1::control/FFYZCCNMCFOI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>Arn:aws:controltower:eu-west-3::control/YLGUUUJTSQLD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>Arn:aws:controltower:sa-east-1::control/UGAKIQSFNZMU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>Arn:aws:controltower:us-west-1::control/IOZSELCXPRJP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SH.NetworkFirewall.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.NetworkFirewall.3</td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Limit network access</td>
<td>• <strong>US East (N. Virginia)</strong></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/DAEZMRPYHWMM</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td>• <strong>US East (Ohio)</strong></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 11.4</td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/ZQAOKRZIJCXXK</td>
</tr>
<tr>
<td></td>
<td>• Limit network access</td>
<td></td>
<td>• <strong>US West (Oregon)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/MNFWRMESSLQI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Canada (Central)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/OVLTUEUQAYRM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Asia Pacific (Sydney)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/TIPAEJGHISPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Asia Pacific (Singapore)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/JVTCW0YSPHTGH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Frankfurt)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/TFSSQYTBVRTD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Ireland)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/JTCGWNUCWXGP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (London)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/SJFAUIKIVZPF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Stockholm)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/OWFEMSEFETJT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Asia Pacific (Mumbai)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/KXECUOXEPWAE</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.NetworkFirewall.4</td>
<td></td>
<td>Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ HGOJKKYHHX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ NTLDJCLFWUL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ CWNUQODQOSL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ NDXZOSZBZAZP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ EVSNHFCZNEP</td>
</tr>
</tbody>
</table>

### SH.NetworkFirewall.4

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.NetworkFirewall.4</td>
<td></td>
<td>Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ HGOJKKYHHX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ NTLDJCLFWUL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ CWNUQODQOSL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ NDXZOSZBZAZP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ EVSNHFCZNEP</td>
</tr>
</tbody>
</table>

- Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/SIHZYNYKWUBD
- Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ECCRBKUVPRPY
- Europe (Paris) arn:aws:controltower:eu-west-3::control/PRDCUMRODGQB
- South America (São Paulo) arn:aws:controltower:saeast-1::control/NBAGBDCLBADT
- US West (N. California) arn:aws:controltower:us-west-1::control/VQSIZTLFDMPV
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Singapore)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/ CERIDGZLWHOZ</td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/ CKQTV0BLJYXU</td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/ IDRAWMSHQVAB</td>
</tr>
<tr>
<td>Europe (London)</td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/ SBJTKQLKUMRB</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/ VFYTJNCARANB</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/ HAH5WPLBYDBY</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/ YXCJHLEGWVHW</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/ LZJFDKKEAPALS</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/ NYONNGGEHZHP</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/ ZTX6GNKAKQSMA</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/ DAOB5SZMEUPW</td>
</tr>
</tbody>
</table>
## SH.NetworkFirewall.5

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.NetworkFirewall.5 | • NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-2  
• PCI DSS version 3.2.1 1.2.1  
• PCI DSS version 3.2.1 1.3  
• PCI DSS version 3.2.1 1.3.1  
• PCI DSS version 3.2.1 1.3.2  
• PCI DSS version 3.2.1 1.3.4  
• PCI DSS version 3.2.1 1.3.6 | • Limit network access | • **US East (N. Virginia)**  
arn:aws:controltower:us-east-1::control/BATNHYBOOGVK  
• **US East (Ohio)**  
arn:aws:controltower:us-east-2::control/CECJUCSKGADN  
• **US West (Oregon)**  
arn:aws:controltower:us-west-2::control/AWRAGIQUYHJDZ  
• **Canada (Central)**  
arn:aws:controltower:ca-central-1::control/UYDVGEXROLLF  
• **Asia Pacific (Sydney)**  
arn:aws:controltower:ap-southeast-2::control/VZMVHIMBIXOW  
• **Asia Pacific (Singapore)**  
arn:aws:controltower:ap-southeast-1::control/XGOFVGCGBCZM  
• **Europe (Frankfurt)**  
arn:aws:controltower:eu-central-1::control/EAJHPCQELMZV  
• **Europe (Ireland)**  
arn:aws:controltower:eu-west-1::control/WWSRLJHIIPWV  
• **Europe (London)**  
arn:aws:controltower:eu-west-2::control/ZENWXCHUOHMG  
• **Europe (Stockholm)**  
arn:aws:controltower:eu-north-1::control/GJZCPVSNOAFF  
• **Asia Pacific (Mumbai)**  
arn:aws:controltower:ap-south-1::control/OWXSVHGINPIX |
### SH.NetworkFirewall.6

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.NetworkFirewall.6</td>
<td>NIST 800-53 Rev 5 AC-4(21)</td>
<td>Limit network access</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/KMTLZUHSLMDH</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/VITJPDKJTMBU</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/SCVMOZZWOZTV</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/MKMOEILVVSHP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/MMCBTVVVBDFM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| - PCI DSS version 3.2.1 1.3.4  
- PCI DSS version 3.2.1 1.3.6 | - Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/GWDDTZUMKSHU  
- Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/BOWRAGMFVVRG  
- Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/ZUUHBYQIZGW5  
- Europe (London)  
arn:aws:controltower:eu-west-2::control/PDJFTMZONXVB  
- Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/LSBUVHESMXCJ  
- Asia Pacific (Mumbai)  
arn:aws:controltower:ap-south-1::control/AYDXHNXYPDCX  
- Asia Pacific (Seoul)  
arn:aws:controltower:ap-northeast-2::control/MUVXUXJSZLBG  
- Asia Pacific (Tokyo)  
arn:aws:controltower:ap-northeast-1::control/LDGXFXHAKIXE  
- Europe (Paris)  
arn:aws:controltower:eu-west-3::control/NDCDQMWSRORR  
- South America (São Paulo)  
arn:aws:controltower:sa-east-1::control/BGLWKRFBNVTN  
- US West (N. California)  
arn:aws:controltower:us-west-1::control/EROQWWAXAUCZ |
## SH.Opensearch.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Opensearch.1</td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Encrypt data at rest</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ IHCAIGKUVWBB</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ HTPRMMEXYQLS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ QOEVDADFSXDN</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ DHTDOXDDWKDT</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ GNNUYUHJXLW</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ ISPPUNAOKJTL</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ ZPPLHOMKJDA</td>
</tr>
<tr>
<td></td>
<td>3.4</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ LJDAJTIFMIBW</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ VRWSVPPXKRH</td>
</tr>
<tr>
<td></td>
<td>8.2.1</td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ SSBMEZ0MBXUD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ UBZMNDYJFFGL</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.Opensearch.2    |           | Limit network access | • US East (N. Virginia)  
                     |           | AC-21   |      |  
                     |           | AC-3    |  
                     |           | AC-3(7) |  
                     |           | AC-4    |  
                     |           | AC-4(21)|  
                     |           | AC-6    |  
                     |           | SC-7    |  
                     |           | SC-7(11)|  
                     |           | SC-7(16)|  
                     |           | SC-7(20)|  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
                     | NIST 800-53 Rev 5 |  
|                    |           |                  | US East (Ohio) arn:aws:controltower:us-east-2::control/NVSQFLJUXNEJ |  
|                    |           |                  | US West (Oregon) arn:aws:controltower:us-west-2::control/QAPHAI0JQJG5 |  
|                    |           |                  | Canada (Central) arn:aws:controltower:ca-central-1::control/VHKUJNWUZHWF |  
|                    |           |                  | Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/QWJLMXTGTTTY |  

SH.Opensearch.2
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/PDIQINMIKBJU</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ELGALPGEXDIO</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/UFAZXZNLUFJM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/FQPSSRYBKFB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/EYYDQBDGWXAG</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/HTJQZ5BBBCWO5</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/NPHMMQWNTQJA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/GKQYHNVCZXUM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/LCPPNYYQSNQC</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/MSRYQGHHAWEWP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/BFCZCVPISWLS</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.Opensearch.3</td>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td>• Encrypt data in transit</td>
<td>• <strong>US East (N. Virginia)</strong>&lt;br&gt;arn:aws:controltower:us-east-1::control/DPIE02VO08BDX</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>• <strong>US East (Ohio)</strong>&lt;br&gt;arn:aws:controltower:us-east-2::control/JXNDKUHIKUPE</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-23</td>
<td></td>
<td>• <strong>US West (Oregon)</strong>&lt;br&gt;arn:aws:controltower:us-west-2::control/MSRORYOIEHFF</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-23(3)</td>
<td></td>
<td>• <strong>Canada (Central)</strong>&lt;br&gt;arn:aws:controltower:ca-central-1::control/VEHXNORMXQFE</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td>• <strong>Asia Pacific (Sydney)</strong>&lt;br&gt;arn:aws:controltower:ap-southeast-2::control/QPRZHOWCFOFW</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td>• <strong>Asia Pacific (Singapore)</strong>&lt;br&gt;arn:aws:controltower:ap-southeast-1::control/IMPWKHACFWKN</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-8(1)</td>
<td></td>
<td>• <strong>Europe (Frankfurt)</strong>&lt;br&gt;arn:aws:controltower:eucentral-1::control/CBWBMTSYADV</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td>• <strong>Europe (Ireland)</strong>&lt;br&gt;arn:aws:controltower:eu-west-1::control/XUAEEQGCKZF</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td>• <strong>Europe (London)</strong>&lt;br&gt;arn:aws:controltower:eu-west-2::control/DAUGSZULKKCD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Stockholm)</strong>&lt;br&gt;arn:aws:controltower:eunorth-1::control/NZTGOXILALHZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Asia Pacific (Mumbai)</strong>&lt;br&gt;arn:aws:controltower:apsouth-1::control/LDDCEQCOMDTZ</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.Opensearch.4</td>
<td>NIST 800-53 Rev 5 AC-2(4)</td>
<td>Establish logging and monitoring</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/GADHHKSJUALK</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4(26)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ONGPXFNQP8SA</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6(9)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/IPGTNQNCEVHE</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/MTRETSLSGME</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/YGRKTWGZGZWP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/GGXZVSAAXJD0</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/MLXUGQDBTLDS</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/OZKIJCNPPBYQ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/UUHIFUJCREJE</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/CIJEXNXYHBM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/YGRKTWGZGZWP</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/VENRTMNHFVDY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/SYCXPGVFVCCL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/SOQCBCP3HEEE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/JPST5WTJEDDK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/KGHPM3JBTZNM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/FPBLTLPITHE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/EZAUSYQJYQDL</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/RPXLWBPBLWOD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/TTTMLSTLIA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/VOVSXKVMKSSY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.7</td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/LIHPYITEEFGH</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.Opensearch.5    | • NIST 800-53 Rev 5 AC-2(4)  
• NIST 800-53 Rev 5 AC-4(26)  
• NIST 800-53 Rev 5 AC-6(9)  
• NIST 800-53 Rev 5 AU-10  
• NIST 800-53 Rev 5 AU-12  
• NIST 800-53 Rev 5 AU-2  
• NIST 800-53 Rev 5 AU-3  
• NIST 800-53 Rev 5 AU-6(3)  
• NIST 800-53 Rev 5 AU-6(4)  
• NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 SC-7(9)  
• NIST 800-53 Rev 5 SI-3(8)  
• NIST 800-53 Rev 5 SI-4(20)  
• NIST 800-53 Rev 5 SI-7(8)  
• PCI DSS version 3.2.1 10.1  
• PCI DSS version 3.2.1 10.2.1  
• PCI DSS version 3.2.1 10.2.2  
• PCI DSS version 3.2.1 10.2.3  
• PCI DSS version 3.2.1 10.2.4  
• PCI DSS version 3.2.1 10.2.5  
• PCI DSS version 3.2.1 10.2.7  
• PCI DSS version 3.2.1 10.3.1 | • Establish logging and monitoring | • US East (N. Virginia)  
ar:aws:controltower:us-east-1::control/  
WAOBYBQZJHTA  
• US East (Ohio)  
ar:aws:controltower:us-east-2::control/  
ABHACZZEEKDD  
• US West (Oregon)  
ar:aws:controltower:us-west-2::control/  
YTI0EVUJX8VO  
• Canada (Central)  
ar:aws:controltower:ca-central-1::control/  
ZWUQUZVGRFGN  
• Asia Pacific (Sydney)  
ar:aws:controltower:ap-southeast-2::control/  
OZIJJZGYJIAUK  
• Asia Pacific (Singapore)  
ar:aws:controltower:ap-southeast-1::control/  
MXRREADODGVF  
• Europe (Frankfurt)  
ar:aws:controltower:eu-central-1::control/  
QAXHUNQCUXSA  
• Europe (Ireland)  
ar:aws:controltower:eu-west-1::control/  
XUQONDGGDGRUR  
• Europe (London)  
ar:aws:controltower:eu-west-2::control/  
FSORDAEQFTUW  
• Europe (Stockholm)  
ar:aws:controltower:eu-north-1::control/  
MCFVQGDHUXSS  
• Asia Pacific (Mumbai)  
ar:aws:controltower:ap-south-1::control/  
GZUMMDQRRYGL |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Opensearch.6</td>
<td></td>
<td></td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Improve availability</td>
<td>11:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Improve availability</td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>14:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>15:</td>
</tr>
</tbody>
</table>

- PCI DSS version 3.2.1 10.3.2
- PCI DSS version 3.2.1 10.3.3
- PCI DSS version 3.2.1 10.3.4
- PCI DSS version 3.2.1 10.3.5
- PCI DSS version 3.2.1 10.3.6

- NIST 800-53 Rev 5 CP-10
- NIST 800-53 Rev 5 CP-6(2)
- NIST 800-53 Rev 5 SC-36
- NIST 800-53 Rev 5 SC-5(2)
- NIST 800-53 Rev 5 SI-13(5)
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/THCUKEEVGVR5</td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/IQ7DVHRNNLNT</td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/HX1YGAHXBFH</td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/KPYLKEFVFLG</td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/NBJRUMLQQPZK</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/AHZDAVZEIKPI</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/NALYWTOTQSQM</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/BRKBMKFMLHEQ</td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/TRTPXUQNQSKE</td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/BYZZFSUVEFIH</td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/WOYZNDSBYFMD</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------------</td>
</tr>
</tbody>
</table>
| SH.Opensearch.7    | • NIST 800-53 Rev 5 AC-2(1)  
• NIST 800-53 Rev 5 AC-3  
• NIST 800-53 Rev 5 AC-3(15)  
• NIST 800-53 Rev 5 AC-3(7)  
• NIST 800-53 Rev 5 AC-5  
• NIST 800-53 Rev 5 AC-6  
• PCI DSS version 3.2.1 7.1.1  
• PCI DSS version 3.2.1 7.2.1  
• PCI DSS version 3.2.1 7.2.2 | • Enforce least privilege | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
TMJKCOOOPZDS  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
HYJEBTQLGRWB  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
XJACA3PWJYEG  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/  
CDTQYPILPGYQ  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
EGTEXCKXTWYU  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/  
YWDFKJTPFDGL  
• Europe (Frankfurt)  
arn:aws:controltower:eucentral-1::control/  
HBOVIWODSIIQ  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/  
QIUOXLURBQXH  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/  
SQMHDHKIDE8FB  
• Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/  
UBSEPWEOOVH  
• Asia Pacific (Mumbai)  
arn:aws:controltower:ap-south-1::control/  
AwwBNORXHGFQ |
### Control API identifiers, by Region

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Opensearch.8</td>
<td>NIST 800-53 Rev 5 AC-17(2)</td>
<td>Encrypt data in transit</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/OEGCJFDHPGEM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/MIUYZGZSCN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 IA-5(1)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/ZYMOYFTGZ00</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/CLTBPMNSISR</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-23</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/MQGZXOCEDKKX</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-23(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-8(1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/JVLECKNIHFJT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/QGKGUEKWKELB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/QHKUXLTYQGOPU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sao-east-1::control/KJJGQDQDNUG</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/KAYPQKVRYEC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/MQGZXOCEDKKX</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

2342
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-8(2)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/YQCKQPEAIJQT</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.3</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 4.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/QBMJTLWYWC9P8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/CCKMAIIIZNSPH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/UBUBNOTQLYHV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/HUPEGIPAJUZG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/YRWIMCSZLLAG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/ELQLURSWUQR8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/SSAMJDSXKUQ8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/EBZSATEEEXTKE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:saeast-1::control/LVIOTWAGDXPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/HETKZGLMYTUT</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.RDS.1           | • NIST 800-53 Rev 5 AC-2(1)  
• NIST 800-53 Rev 5 AC-21  
• NIST 800-53 Rev 5 AC-3  
• NIST 800-53 Rev 5 AC-3(15)  
• NIST 800-53 Rev 5 AC-3(7)  
• NIST 800-53 Rev 5 AC-4  
• NIST 800-53 Rev 5 AC-4(21)  
• NIST 800-53 Rev 5 AC-6  
• NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-2  
• NIST 800-53 Rev 5 CM-2(2)  
• NIST 800-53 Rev 5 CP-10  
• NIST 800-53 Rev 5 CP-6  
• NIST 800-53 Rev 5 CP-6(1)  
• NIST 800-53 Rev 5 CP-6(2)  
• NIST 800-53 Rev 5 CP-9  
• NIST 800-53 Rev 5 SC-36  
• NIST 800-53 Rev 5 SC-5(2)  
• NIST 800-53 Rev 5 SC-7  
• NIST 800-53 Rev 5 SC-7(11)  
• NIST 800-53 Rev 5 SC-7(16)  
| • Enforce least privilege  
• Protect data integrity | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/UWMYPVTQGECU  
• US East (Ohio) arn:aws:controltower:us-east-2::control/OYYKDWBCQTYU  
• US West (Oregon) arn:aws:controltower:us-west-2::control/CAGKPKAGDCTC  
• Canada (Central) arn:aws:controltower:ca-central-1::control/NZXCAQOXXCUB  
• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/KMPUUHKSUHBG  
• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/QZXCSYNMQUwZ  
• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/NCZTIRGKTD7Q  
• Europe (Ireland) arn:aws:controltower:eu-west-1::control/GWPDEHULWHVB  
• Europe (London) arn:aws:controltower:eu-west-2::control/ELQQDOSZOPYE  
• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/RONREVDJLHJ  
• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/EHMNWTQNUQFX |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 11.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 6.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Asia Pacific (Seoul)**
  arn:aws:controltower:ap-northeast-2::control/JZXYYUXYDVES

- **Asia Pacific (Tokyo)**
  arn:aws:controltower:ap-northeast-1::control/TOXGHRQ0OZM

- **Europe (Paris)**
  arn:aws:controltower:europe-west-3::control/BJULHHB3YPSZ

- **South America (São Paulo)**
  arn:aws:controltower:south-america-east-1::control/DRLOIMLUYCCX

- **US West (N. California)**
  arn:aws:controltower:us-west-1::control/EFSULLGMVDIB
### SH.RDS.10

<table>
<thead>
<tr>
<th>Control Identifier</th>
<th>Framework</th>
<th>Control Objective</th>
<th>Control API Identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.RDS.10</td>
<td>NIST 800-53 Rev 5 AC-2(1)</td>
<td>Use strong authentication</td>
<td>US East (N. Virginia) &lt;br&gt;arn:aws:controltower:us-east-1::control/QK6WQNGCDPYW</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>US East (Ohio) &lt;br&gt;arn:aws:controltower:us-east-2::control/JRCVWZHQQKOW</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>US West (Oregon) &lt;br&gt;arn:aws:controltower:us-west-2::control/EFAPZPONOBQN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>Canada (Central) &lt;br&gt;arn:aws:controltower:ca-central-1::control/PRHACJRAZPNL</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td>Asia Pacific (Sydney) &lt;br&gt;arn:aws:controltower:ap-southeast-2::control/SMAVTVYJNJT</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td>Asia Pacific (Singapore) &lt;br&gt;arn:aws:controltower:ap-southeast-1::control/JUHUNMEAVSXZ</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td>Europe (Frankfurt) &lt;br&gt;arn:aws:controltower:eu-central-1::control/SXLQIDOEFSRI</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 8.7</td>
<td></td>
<td>Europe (Ireland) &lt;br&gt;arn:aws:controltower:eu-west-1::control/YPEONMSLBPTL</td>
</tr>
</tbody>
</table>
|                    | PCI DSS version 3.2.1 8.7 | | Europe (London) <br>arn:aws:controltower:eu-
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.RDS.11</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **NIST 800-53 Rev 5 CP-10**
- **NIST 800-53 Rev 5 CP-6**
- **NIST 800-53 Rev 5 CP-6(1)**
- **NIST 800-53 Rev 5 CP-6(2)**
- **NIST 800-53 Rev 5 CP-9**
- **Improve resiliency**

- **US East (N. Virginia)**
  - arn:aws:controltower:us-east-1::control/IFMREHTYREKZ

- **US East (Ohio)**
  - arn:aws:controltower:us-east-2::control/JXWBLIPXTXFM

- **US West (Oregon)**
  - arn:aws:controltower:us-west-2::control/2347
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td></td>
<td>west-2::control/EXXRISQZELSP</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-12</td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/TPWBDXCLLFXK</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/PVJEIOWFEXGW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.7</td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/JKXCQSHKOFWD</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.1</td>
<td></td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/EXPWZLGXOBNE</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/NKIMJVTDDZKZ</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/PUGHZSTXSXHE</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ZHUFJUOHQHAK</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/YITLGYAIRUBP</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/WNGCEPODSWMA</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/UEYPQELH26CQ</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:eu-west-3::control/NOXEGPLREHN</td>
</tr>
<tr>
<td>•</td>
<td></td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:sa-</td>
</tr>
</tbody>
</table>

2348
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.RDS.17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>• Protect configurations</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/DTZXJLHKFLEJ</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/TUGTHWRGHGVM</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CM-2(2)</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/Q2350</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/DTZXJLHKFLEJ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.4</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/TUGTHWRGHGVM</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------</td>
<td>-------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>west-2::control/</td>
<td></td>
<td></td>
<td>THIMEGDOQOLQT</td>
</tr>
<tr>
<td></td>
<td>Canada (Central)</td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td>central-1::control/</td>
<td></td>
<td>AUGOGJCNNLCH</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Sydney)</td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td>southeast-2::control/</td>
<td></td>
<td>SRFNCNDSXZNT</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Singapore)</td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td>southeast-1::control/</td>
<td></td>
<td>SRFHGCHNOQL</td>
</tr>
<tr>
<td></td>
<td>Europe (Frankfurt)</td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:eu-</td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/</td>
</tr>
<tr>
<td></td>
<td>central-1::control/</td>
<td></td>
<td>DJZAPEFIHNFU</td>
</tr>
<tr>
<td></td>
<td>Europe (Ireland)</td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:eu-</td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/</td>
</tr>
<tr>
<td></td>
<td>west-1::control/</td>
<td></td>
<td>BFNXHULOVVIP</td>
</tr>
<tr>
<td></td>
<td>Europe (London)</td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:eu-</td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/</td>
</tr>
<tr>
<td></td>
<td>west-2::control/</td>
<td></td>
<td>ZCZUYDTUHIOI</td>
</tr>
<tr>
<td></td>
<td>Europe (Stockholm)</td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:eu-</td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/</td>
</tr>
<tr>
<td></td>
<td>north-1::control/</td>
<td></td>
<td>FWIVEWQQXNHK</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/</td>
</tr>
<tr>
<td></td>
<td>south-1::control/</td>
<td></td>
<td>YVFDAUWENFAA</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td>northeast-2::control/</td>
<td></td>
<td>EPTKMHACQELH</td>
</tr>
<tr>
<td></td>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:ap-</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/</td>
</tr>
<tr>
<td></td>
<td>northeast-1::control/</td>
<td></td>
<td>DVLKJXCOMXM</td>
</tr>
<tr>
<td></td>
<td>Europe (Paris)</td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:eu-</td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/</td>
</tr>
<tr>
<td></td>
<td>west-3::control/</td>
<td></td>
<td>BZZSMKNBILCF</td>
</tr>
<tr>
<td></td>
<td>South America (São Paulo)</td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:sa-</td>
<td></td>
<td>arn:aws:controltower:sa-</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.RDS.18</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
|                    | • NIST 800-53 Rev 5 AC-4 | • Limit network access | • US East (N. Virginia)  
ar:aws:controltower:us-  
east-1::control/PXYEQHPGSAU  
• US East (Ohio)  
ar:aws:controltower:us-  
east-2::control/JYLXQUQOFUHZ  
• US West (Oregon)  
ar:aws:controltower:us-  
west-2::control/QFDDMLBXZTFMD  
• Canada (Central)  
ar:aws:controltower:ca-  
central-1::control/WGELDILYYVXL  
• Asia Pacific (Sydney)  
ar:aws:controltower:ap-  
southeast-2::control/RBYPDKTCHCVN  
• Asia Pacific (Singapore)  
ar:aws:controltower:ap-  
southeast-1::control/FLLXFLHIFSMQ  
• Europe (Frankfurt)  
ar:aws:controltower:eucen-  
tral-1::control/BPCAMESMWABH  
• Europe (Ireland)  
ar:aws:controltower:euw-  
west-1::control/PSTYKGKBDUTG  
• Europe (London)  
ar:aws:controltower:eul-

2352
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>west-2::control/ZZHOC2FFHRAS</td>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/EJNZYMOZMYKX</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/WUNIHEUEMUJR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/NNRQJBASRGFM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/JSNSKZQFYVBT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/SWEQGXMIMHXT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/HYSDYMKJUVK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/PTTJFTDKIAFZ</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SH.RDS.19**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.RDS.19</td>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td>• Prepare for incident response</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/NXPQTLYAEZFF</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/MHCRBPUNLAM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 11.5</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/2353</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>west-2::control/LRBSFHOKMEYK</td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/DUYBROXNVRZJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/FDBSHSSULULV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/WYTBHZZBIKS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eucentral-1::control/SDCINGKZSOQU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:euwest-1::control/FFWQ0TMRZLCX</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (London) arn:aws:controltower:euwest-2::control/WJYRSJKCXTX</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eunorth-1::control/LAZJDBKFEJMA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:apsouth-1::control/FNYIMXTF0DXS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:apnortheast-2::control/JYTKROMQCXV</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:apnortheast-1::control/CTNWHYDIRSVT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:euwest-3::control/PWEQIFVKNZUB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## SH.RDS.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.RDS.2</td>
<td>• NIST 800-53 Rev 5 AC-21</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>arn:aws:controltower:us-east-1::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>WDBV1IVSLRNF</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>• US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
<td></td>
<td>arn:aws:controltower:us-east-2::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>JOYEALSGIFL</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td>• US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>arn:aws:controltower:us-west-2::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>MZHXJEYNYHPM</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>• Canada (Central)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>LVTHEOBWUSHXI</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>• Asia Pacific (Sydney)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>IWPFNXTZXADU</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td>FXNNEWSUUKUT</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MONNJZHUXUZJV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ZNRDFQKQVSBM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## SH.RDS.20

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.RDS.20          | • NIST 800-53 Rev 5 CA-7  
                    • NIST 800-53 Rev 5 SI-2  
                    • PCI DSS version 3.2.1 11.5 | • Prepare for incident response | • **US East (N. Virginia)**  
arn:aws:controltower:us-east-1::control/BZKHLGWSEYGS  
• **US East (Ohio)**  
arn:aws:controltower:us-east-2::control/QKBXURIFEDNR  
• **US West (Oregon)**  
arn:aws:aws:controltower:us-west-2::control/EZKYVDMHEXTT  
• **Canada (Central)**  
arn:aws:controltower:ca-central-1::control/OODDUWOHHNMJ  
• **Asia Pacific (Sydney)**  
arn:aws:controltower:ap-southeast-2::control/SIZRZVALISEG  
• **Asia Pacific (Singapore)**  
arn:aws:controltower:ap-southeast-1::control/QMPWTHDYXPNF  
• **Europe (Frankfurt)**  
arn:aws:controltower:eu-central-1::control/ MGGSWFHNHYEYW  
• **Europe (Ireland)**  
arn:aws:controltower:eu-west-1::control/OOZTVTPLFUZE  
• **Europe (London)**  
arn:aws:controltower:eu-west-2::control/SCDHHEPMDRZYD  
• **Europe (Stockholm)**  
arn:aws:controltower:eu-north-1::control/AYTEEMREPGTY  
• **Asia Pacific (Mumbai)**  
arn:aws:controltower:ap-south-1::control/NVFJXDYXEZUI |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.RDS.21</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
|                    | • NIST 800-53 Rev 5 CA-7 | • Prepare for incident response | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/LSOCQKOMTNUV  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/HXSMRUIUXHEQL  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/VNVOSKGSBKYM  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/JUUCAVNVKIFZ  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/BEAJMSSHPLZAF |

• Asia Pacific (Seoul)  
arn:aws:controltower:ap-northeast-2::control/HVLENLXSMYVM  
• Asia Pacific (Tokyo)  
arn:aws:controltower:ap-northeast-1::control/YMSJKTCPZVLQ  
• Europe (Paris)  
arn:aws:controltower:eu-west-3::control/JWYATNKBGECV  
• South America (São Paulo)  
arn:aws:controltower:sa-east-1::control/OAIMBCJMFKMH  
• US West (N. California)  
arn:aws:controltower:us-west-1::control/ZWIFHLDJHCYK
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/ SFJABILZRZSE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/ BDSWAPWAE300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/ KFWXGBPPKFLG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ IREBVNNCJFQN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ XZAJRQOSAI5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ MUDGOLDWJQFJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ GZCLBNPOKRNZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ YXJRNQXFFB7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ UEMPPORPOEXD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ ONKVOCAH0MG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ GTXGBGCVGHQI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.RDS.22          | • NIST 800-53 Rev 5 CA-7  
   • NIST 800-53 Rev 5 SI-2  
   • PCI DSS version 3.2.1 11.5 | • Prepare for incident response | • US East (N. Virginia)  
   arn:aws:controltower:us-east-1::control/  
   APIPOBXRV0AY  
   • US East (Ohio)  
   arn:aws:controltower:us-east-2::control/  
   XNOPTYHHXQYF  
   • US West (Oregon)  
   arn:aws:controltower:us-west-2::control/  
   YNHFAJNIYTDT  
   • Canada (Central)  
   arn:aws:controltower:ca-central-1::control/  
   ZGGHJOYOXRWY  
   • Asia Pacific (Sydney)  
   arn:aws:controltower:ap-southeast-2::control/  
   HGUANBQWZZT  
   • Asia Pacific (Singapore)  
   arn:aws:controltower:ap-southeast-1::control/  
   EQAAWXJPTMKT  
   • Europe (Frankfurt)  
   arn:aws:controltower:eu-central-1::control/  
   NOQNKAMBARQQ  
   • Europe (Ireland)  
   arn:aws:controltower:eu-west-1::control/  
   FHEENQFCWRW  
   • Europe (London)  
   arn:aws:controltower:eu-west-2::control/  
   USADZKONGEWL  
   • Europe (Stockholm)  
   arn:aws:controltower:eu-north-1::control/  
   WCCGGPOESKUN  
   • Asia Pacific (Mumbai)  
   arn:aws:controltower:ap-south-1::control/  
   QUVLHKDDEGEY |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.RDS.23</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Control identifier**

- SH.RDS.23

**Framework**

- NIST 800-53 Rev 5 AC-4
- NIST 800-53 Rev 5 AC-4(21)
- NIST 800-53 Rev 5 SC-7
- NIST 800-53 Rev 5 SC-7(11)
- NIST 800-53 Rev 5 SC-7(16)
- NIST 800-53 Rev 5 SC-7(21)
- NIST 800-53 Rev 5 SC-7(4)
- NIST 800-53 Rev 5 SC-7(5)
- PCI DSS version 3.2.1 1.2.1
- PCI DSS version 3.2.1 1.3.1

**Control objective**

- Limit network access

**Control API identifiers, by Region**

- **Asia Pacific (Seoul)**
  arn:aws:controltower:ap-northeast-2::control/XGNIROGTVPW

- **Asia Pacific (Tokyo)**
  arn:aws:controltower:ap-northeast-1::control/EXEHMGV0GPEL

- **Europe (Paris)**
  arn:aws:controltower:eu-west-3::control/MEMSFTMAYJCK

- **South America (São Paulo)**
  arn:aws:controltower:sa-east-1::control/ZIKCBMBK0YQG

- **US West (N. California)**
  arn:aws:controltower:us-west-1::control/QMZEDTMIMIZLU

- **US East (N. Virginia)**
  arn:aws:controltower:us-east-1::control/YMXSTCEYXBPV

- **US East (Ohio)**
  arn:aws:controltower:us-east-2::control/BMHOKULQGTAT

- **US West (Oregon)**
  arn:aws:controltower:us-west-2::control/TJALNCJQYRFH

- **Canada (Central)**
  arn:aws:controltower:ca-central-1::control/WSSFKIXFVIRY

- **Asia Pacific (Sydney)**
  arn:aws:controltower:ap-southeast-2::control/ZRQKSDQENQGQ
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td>• Asia Pacific (Singapore)  arn:aws:controltower:ap-southeast-1::control/QXHTVIKMMTW</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td>• Europe (Frankfurt)  arn:aws:controltower:eu-central-1::control/VXAKLAUEPMU</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• Europe (Ireland)  arn:aws:controltower:eu-west-1::control/RASOHQDBGUM</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• Europe (London)  arn:aws:controltower:eu-west-2::control/PLKDXQDDBCM</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• Europe (Stockholm)  arn:aws:controltower:eu-north-1::control/IVCTUYVNYKZS</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• Asia Pacific (Mumbai)  arn:aws:controltower:ap-south-1::control/FAFQHCERANWF</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• Asia Pacific (Seoul)  arn:aws:controltower:ap-northeast-2::control/VIWCZSTJLLWW</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• Asia Pacific (Tokyo)  arn:aws:controltower:ap-northeast-1::control/HIWFHIFSZSKM</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• Europe (Paris)  arn:aws:controltower:eu-west-3::control/XNDQOTSALSLX</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• South America (São Paulo)  arn:aws:controltower:sa-east-1::control/WGAKCUZZVTHP</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• US West (N. California)  arn:aws:controltower:us-west-1::control/YMDPRQFAXKCO</td>
</tr>
</tbody>
</table>

2362
## SH.RDS.25

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.RDS.25          | • NIST 800-53 Rev 5 CA-9(1)  
                    • NIST 800-53 Rev 5 CM-2  
                    • PCI DSS version 3.2.1 2.1 | • Protect configurations | • **US East (N. Virginia)**
|                    |           |                   | arn:aws:controltower:us-east-1::control/BBSLGPRKDCNJ |
|                    |           |                   | • **US East (Ohio)**
|                    |           |                   | arn:aws:controltower:us-east-2::control/RLYGPPANIDPZ |
|                    |           |                   | • **US West (Oregon)**
|                    |           |                   | arn:aws:aws:controltower:us-west-2::control/QCWMEMJPWTVG |
|                    |           |                   | • **Canada (Central)**
|                    |           |                   | arn:aws:controltower:ca-central-1::control/HKYZMPYFMAFO |
|                    |           |                   | • **Asia Pacific (Sydney)**
|                    |           |                   | arn:aws:controltower:ap-southeast-2::control/BGPIYTQ2CFXL |
|                    |           |                   | • **Asia Pacific (Singapore)**
|                    |           |                   | arn:aws:controltower:ap-southeast-1::control/NDCFQRXZYLGR |
|                    |           |                   | • **Europe (Frankfurt)**
|                    |           |                   | arn:aws:controltower:eu-central-1::control/GROVJRZIBJCL |
|                    |           |                   | • **Europe (Ireland)**
|                    |           |                   | arn:aws:controltower:eu-west-1::control/MZHTGUBEWDXJ |
|                    |           |                   | • **Europe (London)**
|                    |           |                   | arn:aws:controltower:eu-west-2::control/RVSFMXXOBZS |
|                    |           |                   | • **Europe (Stockholm)**
|                    |           |                   | arn:aws:controltower:eu-north-1::control/ONRGZMZOUBUFB |
|                    |           |                   | • **Asia Pacific (Mumbai)**
|                    |           |                   | arn:aws:controltower:ap-south-1::control/AWJUNDMLTRRD |
### SH.RDS.27

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.RDS.27          |           | Encrypt data at rest | • US East (N. Virginia)  
ar:aws:controltower:us-east-1::control/  
DROQPQLJBTES  
• US East (Ohio)  
ar:aws:controltower:us-east-2::control/  
YOTEP0ZWWVTA  
• US West (Oregon)  
ar:aws:controltower:us-west-2::control/  
AEXJPTLILJTV  
• Canada (Central)  
ar:aws:controltower:ca-central-1::control/  
HYHMK3XEHP5  
• Asia Pacific (Sydney)  
ar:aws:controltower:ap-southeast-2::control/  
YIQWYGANUPYO  
• PCI DSS version 3.2.1  
3.1  
• PCI DSS version 3.2.1  
3.4  
• PCI DSS version 3.2.1  
8.2.1  
• NIST 800-53 Rev 5  
CA-9(1)  
• NIST 800-53 Rev 5  
CM-3(6)  
• NIST 800-53 Rev 5  
SC-13  
• NIST 800-53 Rev 5  
SC-28  
• NIST 800-53 Rev 5  
SC-28(1)  
• NIST 800-53 Rev 5  
SC-7(10)  
• NIST 800-53 Rev 5  
SI-7(6)  
• Encrypt data at rest |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/UEUHZXQNAEBCQ</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/YHFDAYZPEVWO</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/SMQYFRRNIEJB</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/TZIPAEKMFTMH</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/SHKEUJZRGZDP</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/VLLLLYEMUGOC</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/ONFIOHPFSHGK</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/DRFBXVBNCRS</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/DJZYDUIVJLQI</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/WAAVLKURPRRQ</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/WCSGBVJYK3KB</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.RDS.3</td>
<td>• CIS AWS Benchmark 1.4 2.3.1</td>
<td>Encrypt data at rest</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ VGWJZSHSYIBM</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ VCGAQSCJ3VXR</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ NMQXECEAQCBC</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ KAZMY3AUZYSY</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ OKWMNXY2RUVG</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/ DUPIMTBMZJBD</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(10)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/ KNPNIWDEVEJG</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/ HBTSVIZWIAQY</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/ ZQCGVSRDNLZEK</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/ QGGSRCNFPB0N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/ QLBPHXRQYFTE</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.RDS.4           |           | Encrypt data at rest | • US East (N. Virginia) 
arn:aws:controltower:us-east-1::control/
BZBLCONTBAMD  
• US East (Ohio)    
arn:aws:controltower:us-east-2::control/
JFTFUHKRVKVS   
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/
0GAIJABSGBT  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/
VXU0LFZ3HRXA   
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/
U0CQAV0kBTP    |

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.RDS.4           |           | Encrypt data at rest | • Asia Pacific (Seoul) 
arn:aws:controltower:ap-northeast-2::control/
VMJ0VSBQHNL      
• Asia Pacific (Tokyo) 
arn:aws:controltower:ap-northeast-1::control/
ICYIAUGUTWUB     
• Europe (Paris)    
arn:aws:controltower:eu-west-3::control/
ABSIYZAIAWCN     
• South America (São Paulo) 
arn:aws:controltower:sa-east-1::control/
WWDYZRWIDTXW     
• US West (N. California) 
arn:aws:controltower:us-west-1::control/
TGXCMCDKMXP      |

• NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-3(6)  
• NIST 800-53 Rev 5 SC-13  
• NIST 800-53 Rev 5 SC-28  
• NIST 800-53 Rev 5 SC-28(1)  
• NIST 800-53 Rev 5 SC-7(10)  
• NIST 800-53 Rev 5 SI-7(6)  
• PCI DSS version 3.2.1 3.4  
• PCI DSS version 3.2.1 8.2.1
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/ SWWDMMPJTVUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/ KZAQFIPMORTE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/ LREEKQPJSYDJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ CZGUUSLYJXEQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ BBVSYFCDKBUZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ DACICEFQAULS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ TXZGWJTMYAS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ CLRBIHMGNH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ BKZVWTBCYGUF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ RZHJRUQRPWRV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ NOZHCDEDTOTS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## SH.RDS.5

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.RDS.5</td>
<td>NIST 800-53 Rev 5 CP-10</td>
<td>Improve availability</td>
<td>US East (N. Virginia) (\text{arn:aws:controltower:us-east-1::control/ WAIĐVEUJUPWI})</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td>US East (Ohio) (\text{arn:aws:controltower:us-east-2::control/ NTFDQTWLGZW0})</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-36</td>
<td></td>
<td>US West (Oregon) (\text{arn:aws:controltower:us-west-2::control/ WQRKCNCGXMEZ})</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>Canada (Central) (\text{arn:aws:controltower:ca-central-1::control/ XMP0BIWBCBI})</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td>Asia Pacific (Sydney) (\text{arn:aws:controltower:ap-southeast-2::control/ IRS1BR2RNPGJ})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Singapore) (\text{arn:aws:controltower:ap-southeast-1::control/ NNYNUMCTQZCZ})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Frankfurt) (\text{arn:aws:controltower:europe-central-1::control/ RDRYY1DEWNY})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Ireland) (\text{arn:aws:controltower:europe-west-1::control/ ABTYUCPXKSGJ})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (London) (\text{arn:aws:controltower:europe-west-2::control/ HKTFGMWLCXVS})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) (\text{arn:aws:controltower:europe-north-1::control/ YREOQGNQKBLJ})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) (\text{arn:aws:controltower:ap-south-1::control/ GPUFSACIRWXI})</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.RDS.6           |           |                  | • Asia Pacific (Seoul)  
arn:aws:controltower:ap-northeast-2::control/QSPRBKYKNWKM  
• Asia Pacific (Tokyo)  
arn:aws:controltower:ap-northeast-1::control/OTHPZRZHRYNM  
• Europe (Paris)  
arn:aws:controltower:eu-west-3::control/SEMODNJPIZDI  
• South America (São Paulo)  
arn:aws:controltower:sa-east-1::control/HDDRWBVHAVWK  
• US West (N. California)  
arn:aws:controltower:us-west-1::control/IXXVGFGVMRZK |

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.RDS.6           |           |                  | • NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 SI-2  
• Establish logging and monitoring  
arn:aws:controltower:us-east-1::control/CQFNSAHGKXYR  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/QJCUZIBGHFIW  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/EYAIAPJSQWR  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/DKSRZCX3NFHQ  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/LJQFIHVOJEWC |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/WGVCYIMZTDK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/JKBKECQARADY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/OEPIOAXLVJQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/FUVWSPBRWMJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/OITUKLXALQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/VHPWTRYBF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/MEYNQLIBFVH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/BKCRQHWT0IDQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/XOMMAAMDJJSN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:saeast-1::control/KLAQDLQDNQCO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/EAZJCJVTNTTK</td>
</tr>
</tbody>
</table>
## SH.RDS.8

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.RDS.8           | • NIST 800-53 Rev 5 CA-9(1)  
|                    | • NIST 800-53 Rev 5 CM-2  
|                    | • NIST 800-53 Rev 5 CM-2(2)  
|                    | • NIST 800-53 Rev 5 CM-3  
|                    | • NIST 800-53 Rev 5 SC-5(2)  
|                    | • PCI DSS version 3.2.1 2.2 | • Improve availability | • **US East (N. Virginia)**  
|                    |                                      |                                 | arn:aws:controltower:us-east-1::control/OHXOGCEUBHBF |
|                    |                                      |                                 | • **US East (Ohio)**  
|                    |                                      |                                 | arn:aws:controltower:us-east-2::control/CCQXNXARGJAR |
|                    |                                      |                                 | • **US West (Oregon)**  
|                    |                                      |                                 | arn:aws:aws:controltower:us-west-2::control/EZGQFHTMBAWO |
|                    |                                      |                                 | • **Canada (Central)**  
|                    |                                      |                                 | arn:aws:controltower:ca-central-1::control/WFLIUEPSIAYX |
|                    |                                      |                                 | • **Asia Pacific (Sydney)**  
|                    |                                      |                                 | arn:aws:controltower:ap-southeast-2::control/CDKCDVDPAZJE |
|                    |                                      |                                 | • **Asia Pacific (Singapore)**  
|                    |                                      |                                 | arn:aws:controltower:ap-southeast-1::control/GINLEIXHNSYL |
|                    |                                      |                                 | • **Europe (Frankfurt)**  
|                    |                                      |                                 | arn:aws:controltower:eu-central-1::control/SQHTJKLMQJGB |
|                    |                                      |                                 | • **Europe (Ireland)**  
|                    |                                      |                                 | arn:aws:controltower:eu-west-1::control/XBIZWFFXCCON |
|                    |                                      |                                 | • **Europe (London)**  
|                    |                                      |                                 | arn:aws:controltower:eu-west-2::control/VQMCVFHXFAQQ |
|                    |                                      |                                 | • **Europe (Stockholm)**  
|                    |                                      |                                 | arn:aws:controltower:eu-north-1::control/NMWOPNACHOL |
|                    |                                      |                                 | • **Asia Pacific (Mumbai)**  
|                    |                                      |                                 | arn:aws:controltower:ap-south-1::control/LWZROQWERBHQG |

2372
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4(26)</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/OTXQBSPGPGYU</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6(9)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/DBSDVNSHWIOG</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/RYUXCMWBRCAAD</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/FPDDNGIQPZQT</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/TBIMPMPTMBJTL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/GXZQLMFFQWIDW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/IIFLSTVTRDZG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/KMLDOBVZKCT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/REJGKLYPHVDN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/KAQQDXQFJYRU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/PSHCDLXAFPIZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/EIZJBSDXPFGK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/HNXTEFHVWUVU</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/NDKTXPFKJX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/IXAUVMTYOJTU</td>
</tr>
</tbody>
</table>
## SH.Redshift.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Redshift.1</td>
<td>• NIST 800-53 Rev 5 AC-21</td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/QGVYRDNRLLPZ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/VRMIAJEHXVEF</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/KHOSQGUZDTNH</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/RFNNTQDNNSXM</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/WVIYUKDFONY</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AWCJAMQENDQO</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/IYGOKELWJWMB</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/CCXNECVPVRLH</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/RHXEMVOQGMNY</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/LZKIQTREDTFY</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/WXXRGRFXMKHZ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SH.Redshift.10

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.Redshift.10     |           | Encrypt data at rest | • US East (N. Virginia) | arn:aws:controltower:us-east-1::control/NNMBPYPYCIYIAW  
|                    | PCI DSS version 3.2.1 1.3.1 |                     | • US East (Ohio) | arn:aws:controltower:us-east-2::control/FGDKUQO3DCQG  
|                    | PCI DSS version 3.2.1 1.3.2 |                     | • US West (Oregon) | arn:aws:controltower:us-west-2::control/CLQQXAYJAFAH  
|                    | PCI DSS version 3.2.1 1.3.4 |                     | • Canada (Central) | arn:aws:controltower:ca-central-1::control/SFTAS3FBQOZK  
|                    | PCI DSS version 3.2.1 1.3.6 |                     | • Asia Pacific (Sydney) | arn:aws:controltower:ap-southeast-2::control/LQIBXYSZTOTD  
|                    | PCI DSS version 3.2.1 2.2.2 |                     |                     |  
|                    | PCI DSS version 3.2.1 3.4 |                     |                     |  
|                    | PCI DSS version 3.2.1 3.5.3 |                     |                     |  
|                    | PCI DSS version 3.2.1 8.2.1 |                     |                     |  
|                    | NIST 800-53 Rev 5 CA-9(1) |                     | • US East (N. Virginia) | arn:aws:controltower:us-east-1::control/NNMBPYPYCIYIAW  
|                    | NIST 800-53 Rev 5 CM-3(6) |                     | • US East (Ohio) | arn:aws:controltower:us-east-2::control/FGDKUQO3DCQG  
|                    | NIST 800-53 Rev 5 SC-28 |                     | • Canada (Central) | arn:aws:controltower:ca-central-1::control/SFTAS3FBQOZK  
|                    | NIST 800-53 Rev 5 SC-28(1) |                     | • Asia Pacific (Sydney) | arn:aws:controltower:ap-southeast-2::control/LQIBXYSZTOTD  
|                    | NIST 800-53 Rev 5 SI-7(6) |                     |                     |  
|                    | PCI DSS version 3.2.1 3.4 |                     |                     |  
|                    | PCI DSS version 3.2.1 3.5.3 |                     |                     |  
|                    | PCI DSS version 3.2.1 8.2.1 |                     |                     |  
|                    | PCI DSS version 3.2.1 1.3.1 |                     |                     |  
|                    | PCI DSS version 3.2.1 1.3.2 |                     |                     |  
|                    | PCI DSS version 3.2.1 1.3.4 |                     |                     |  
|                    | PCI DSS version 3.2.1 1.3.6 |                     |                     |  
|                    | PCI DSS version 3.2.1 2.2.2 |                     |                     |  
|                    | PCI DSS version 3.2.1 3.4 |                     |                     |  
|                    | PCI DSS version 3.2.1 3.5.3 |                     |                     |  
|                    | PCI DSS version 3.2.1 8.2.1 |                     |                     |  
|                    | PCI DSS version 3.2.1 1.3.1 |                     |                     |  
|                    | PCI DSS version 3.2.1 1.3.2 |                     |                     |  
|                    | PCI DSS version 3.2.1 1.3.4 |                     |                     |  
|                    | PCI DSS version 3.2.1 1.3.6 |                     |                     |  
|                    | PCI DSS version 3.2.1 2.2.2 |                     |                     |  
|                    | PCI DSS version 3.2.1 3.4 |                     |                     |  
|                    | PCI DSS version 3.2.1 3.5.3 |                     |                     |  
|                    | PCI DSS version 3.2.1 8.2.1 |                     |                     |  

- **Asia Pacific (Seoul)** arn:aws:controltower:ap-northeast-2::control/KUKRAWPCWPD
- **Asia Pacific (Tokyo)** arn:aws:controltower:ap-northeast-1::control/AYTBPCVDXBOV
- **Europe (Paris)** arn:aws:controltower:eu-west-3::control/XJVAUJAVHWDX
- **South America (São Paulo)** arn:aws:controltower:sa-east-1::control/YQJNEMQ5MPCG
- **US West (N. California)** arn:aws:controltower:us-west-1::control/WUADYLMPFVGP
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/FE00FZLBVNOD</td>
<td></td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/APZBNDSLBEGY</td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/TZZYEMVXIONE</td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ZVKXZTUNFLSZ</td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/LUUIEVUKWMHU</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/WJWRWAEUMDCO</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/HLLTIBMTADKA</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/LYYELPWNRVL</td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/TSPTYOGXSED</td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/KAA8MIDGCLBK</td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ENSMMXOHZMQJ</td>
<td></td>
</tr>
</tbody>
</table>
## SH.Redshift.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.Redshift.2      | • NIST 800-53 Rev 5 AC-4  
                    • NIST 800-53 Rev 5 SC-13  
                    • NIST 800-53 Rev 5 SC-23  
                    • NIST 800-53 Rev 5 SC-23(3)  
                    • NIST 800-53 Rev 5 SC-7(4)  
                    • NIST 800-53 Rev 5 SC-8  
                    • NIST 800-53 Rev 5 SC-8(1)  
                    • NIST 800-53 Rev 5 SC-8(2)  
                    • PCI DSS version 3.2.1 2.3  
                    • PCI DSS version 3.2.1 4.1  
                    • PCI DSS version 3.2.1 8.2.1 | • Encrypt data in transit | • **US East (N. Virginia)** arn:aws:controltower:us-east-1::control/GJLYVQTNPXNP  
• **US East (Ohio)** arn:aws:controltower:us-east-2::control/QZALUPN2SSAE  
• **US West (Oregon)** arn:aws:controltower:us-west-2::control/YHXBAMZ2WWSF  
• **Canada (Central)** arn:aws:controltower:ca-central-1::control/YMGCCUBQWEZW  
• **Asia Pacific (Sydney)** arn:aws:controltower:ap-southeast-2::control/ORJWXAPJACW  
• **Asia Pacific (Singapore)** arn:aws:controltower:ap-southeast-1::control/JBOXYUPVWNZ0  
• **Europe (Frankfurt)** arn:aws:controltower:eu-central-1::control/HFS1BFFEDEQY  
• **Europe (Ireland)** arn:aws:controltower:eu-west-1::control/BSBJOQWTSRSX  
• **Europe (London)** arn:aws:controltower:eu-west-2::control/BAZMHRYHVCQG  
• **Europe (Stockholm)** arn:aws:controltower:eu-north-1::control/QXUXFMQMNJNV  
• **Asia Pacific (Mumbai)** arn:aws:controltower:ap-south-1::control/NDBMIUUFYK
## SH.Redshift.4

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Redshift.4</td>
<td>NIST 800-53 Rev 5 AC-2(4)</td>
<td>Establish logging and monitoring</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/UGPCVGGZGVGBN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4(26)</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/VVVOX0KKREWD</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6(9)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/HREVCGLTRGZP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-10</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/JWQMOSWEPUQN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-12</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/UNNOEVFPFBMC</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-2</td>
<td></td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/QINDHCUJDPR</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-3</td>
<td></td>
<td>Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ORADMUNKKIYS</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(3)</td>
<td></td>
<td>Europe (Paris) arn:aws:controltower:europe-west-3::control/YQJYRYKIWLYT</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AU-6(4)</td>
<td></td>
<td>South America (São Paulo) arn:aws:controltower:south-america-east-1::control/PUKFRSLSTCGTP</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CA-7</td>
<td></td>
<td>US West (N. California) arn:aws:controltower:us-west-1::control/MABMUFJZSIUD</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/FZJIXFLBDRT</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/THH5J3D0RTML</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(20)</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-1::control/WXRFQPACPRH</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-7(8)</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/TXRQFGFPRH</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/JHIVYPB00DH</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/VAONGWDMQ3RF</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/BDVLMXDLACBS</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.3</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/DTMFCWBBWBB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.4</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:saeast-1::control/EVHRCYXXFH</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.2.5</td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/OCZIX3TBKZD</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.Redshift.6</td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td>• Manage vulnerabilities</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/TK5UMFLV5LUE</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/JKA4H44IU4ZK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CP-9</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/TP5CF3GMJJSJ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/MWBNTMNYOQBQ</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-2</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/WMXBZ0HRKU0Q</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-2(2)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/DK4ZU3RDLH4</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-2(4)</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/TMJCFUASV0FO</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-2(5)</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/SNCSYNO9QMMH</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 6.2</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/YGTSKVHAGTU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/YRROEF0YJQLY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/WMTYV1K0AEOM</td>
</tr>
</tbody>
</table>
### SH.Redshift.7

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.Redshift.7      |           | • Limit network access | • US East (N. Virginia)  
arwn:aws:controltower:us-east-1::control/EFKEWIIFDJEZ  
• US East (Ohio)   
arwn:aws:controltower:us-east-2::control/LRZEWAVMSHF  
• US West (Oregon)  
arwn:aws:controltower:us-west-2::control/JSLNDFAYMRXF  
• Canada (Central) 
arwn:aws:controltower:ca-central-1::control/VZIZLXFWAEFL  
• Asia Pacific (Sydney)  
arwn:aws:controltower:ap-southeast-2::control/MVNWOYAUMTGJ |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td>Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/RERSLKEIBQHG</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/HOISPCIAYQCB</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/WWFLIKXSJYG</td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/FDQXVYALJBAT</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/NDDIAQWLEPTB</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-2::control/DSKLBSLANWJM</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/CUGFTMBJPKPT</td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/PKUZKCMQQKRR</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.Redshift.8</td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Protect configurations</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• US East (N. Virginia)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:us-east-1::control/DGHUXCTZEDYI</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• US East (Ohio)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:us-east-2::control/BEPNPCQCPDWJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• US West (Oregon)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:us-west-2::control/KLCNBPBVPMMD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Canada (Central)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:ca-central-1::control/OUMGXBPILINOD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Sydney)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:ap-southeast-2::control/XIDWPGTUKALT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/FREYXDZYZTBB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/XFBMDQEICXZM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/ROJQXUSTAPYS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (London)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/DIPEQYZCPMLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/ABEXBGLMXTJB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/LNLPI8KGAUQR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>SH.Redshift.9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Asia Pacific (Seoul)**
  - 
    - 
      - arn:aws:controltower:ap-northeast-2::control/KHSJIQXMKUKN

- **Asia Pacific (Tokyo)**
  - 
    - 
      - arn:aws:controltower:ap-northeast-1::control/HQIOVIJMPDMP

- **Europe (Paris)**
  - 
    - 
      - arn:aws:controltower:eu-west-3::control/ZLWJDNJ0DJN

- **South America (São Paulo)**
  - 
    - 
      - arn:aws:controltower:south-america-east-1::control/XIJAJTJYHCJI

- **US West (N. California)**
  - 
    - 
      - arn:aws:controltower:us-west-1::control/OUYAXYXNBBV

- **NIST 800-53 Rev 5 CA-9(1)**
  - 
    - 
      - Protect configurations

- **NIST 800-53 Rev 5 CM-2**
  - 
    - 

- **PCI DSS version 3.2.1 2.1**
  - 
    - 

- **US East (N. Virginia)**
  - 
    - 
      - arn:aws:controltower:us-east-1::control/SMGEEELNDBKM

- **US East (Ohio)**
  - 
    - 
      - arn:aws:controltower:us-east-2::control/OVOHLXKXBPQL

- **US West (Oregon)**
  - 
    - 
      - arn:aws:controltower:us-west-2::control/UQQKZEWIILTCV

- **Canada (Central)**
  - 
    - 
      - arn:aws:controltower:central-1::control/CS6G13BMPVER

- **Asia Pacific (Sydney)**
  - 
    - 
      - arn:aws:controltower:ap-southeast-2::control/TTPOHLO5ZBPO

---

**SH.Redshift.9**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.Redshift.9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **NIST 800-53 Rev 5 CA-9(1)**
  - 
    - 

- **NIST 800-53 Rev 5 CM-2**
  - 
    - 

- **PCI DSS version 3.2.1 2.1**
  - 
    - 

- **Protect configurations**
  - 

- **US East (N. Virginia)**
  - 
    - 
      - arn:aws:controltower:us-east-1::control/SMGEEELNDBKM

- **US East (Ohio)**
  - 
    - 
      - arn:aws:controltower:us-east-2::control/OVOHLXKXBPQL

- **US West (Oregon)**
  - 
    - 
      - arn:aws:controltower:us-west-2::control/UQQKZEWIILTCV

- **Canada (Central)**
  - 
    - 
      - arn:aws:controltower:central-1::control/CS6G13BMPVER

- **Asia Pacific (Sydney)**
  - 
    - 
      - arn:aws:controltower:ap-southeast-2::control/TTPOHLO5ZBPO

---

2385
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/THQBUSXDS3ML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/TVVIZSSMLNIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/XSRKMHENVRYZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/PMKUADMALEE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/UNTIRUWZWUNN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/DNJVACRRVQMJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/VYGTYLRWZYAD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/VZUUWSEPMGAG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/GQRHHERYSAUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sao-east-1::control/GMLYQLDQHXH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/XFUBODJVVTNX</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## SH.S3.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.S3.1            | • CIS AWS Benchmark 1.4 2.1.5  
                    | • NIST 800-53 Rev 5 AC-2(1)  
                    | • NIST 800-53 Rev 5 AC-21  
                    | • NIST 800-53 Rev 5 AC-3  
                    | • NIST 800-53 Rev 5 AC-3(15)  
                    | • NIST 800-53 Rev 5 AC-3(7)  
                    | • NIST 800-53 Rev 5 AC-4  
                    | • NIST 800-53 Rev 5 AC-4(21)  
                    | • NIST 800-53 Rev 5 AC-6  
                    | • NIST 800-53 Rev 5 AU-9(2)  
                    | • NIST 800-53 Rev 5 CA-7  
                    | • NIST 800-53 Rev 5 CP-10  
                    | • NIST 800-53 Rev 5 CP-6(2)  
                    | • NIST 800-53 Rev 5 CP-9  
                    | • NIST 800-53 Rev 5 SC-5(2)  
                    | • NIST 800-53 Rev 5 SC-7  
                    | • NIST 800-53 Rev 5 SC-7(11)  
                    | • NIST 800-53 Rev 5 SC-7(16)  
                    | • NIST 800-53 Rev 5 SC-7(20)  
                    | • NIST 800-53 Rev 5 SC-7(21)  
                    | • NIST 800-53 Rev 5 SC-7(3)  
                    | • NIST 800-53 Rev 5 SC-7(4)  
|                    | • Enforce least privilege | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/XKHRQYFBBHP  
|                    |                        | • US East (Ohio) arn:aws:controltower:us-east-2::control/JRCVWAZSRAPY  
|                    |                        | • US West (Oregon) arn:aws:controltower:us-west-2::control/PSJEDKJBNWR  
|                    |                        | • Canada (Central) arn:aws:controltower:ca-central-1::control/ARAAXROCJLMK  
|                    |                        | • Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/UBQSERAFQQLI  
|                    |                        | • Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/TFBJICHQSOQCX  
|                    |                        | • Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/UJQJNPZSGLTA  
|                    |                        | • Europe (Ireland) arn:aws:controltower:eu-west-1::control/LSUJTPOECECR  
|                    |                        | • Europe (London) arn:aws:controltower:eu-west-2::control/BFOHYIZSLOKR  
|                    |                        | • Europe (Stockholm) arn:aws:controltower:eu-north-1::control/DHADSFAEUWZS  
|                    |                        | • Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/DDFXZZZGUALP  

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-3(8)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-4(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 11.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 7.2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Asia Pacific (Seoul)
  arn:aws:controltower:ap-northeast-2::control/AGLATCNIJPRQ

• Asia Pacific (Tokyo)
  arn:aws:controltower:ap-northeast-1::control/VRH0XLIG00YV

• Europe (Paris)
  arn:aws:controltower:europe-west-3::control/ECBYLUUJDHST

• South America (São Paulo)
  arn:aws:controltower:south-america-east-1::control/EYOPEGKZSTTF

• US West (N. California)
  arn:aws:controltower:us-west-1::control/XWNNUO0STDRE
## SH.S3.10

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.S3.10           | • NIST 800-53 Rev 5 AU-9(2)  
• NIST 800-53 Rev 5 CP-10  
• NIST 800-53 Rev 5 CP-6(2)  
• NIST 800-53 Rev 5 CP-9  
• NIST 800-53 Rev 5 SC-5(2)  
• NIST 800-53 Rev 5 SI-13(5)  
• PCI DSS version 3.2.1 10.5.3  
• PCI DSS version 3.2.1 10.5.4  
• PCI DSS version 3.2.1 10.7  
• PCI DSS version 3.2.1 3.1 | • Optimize costs  
• Improve availability | • **US East (N. Virginia)**  
arn:aws:controltower:us-east-1::control/YXSHBLNRJEIL  
• **US East (Ohio)**  
arn:aws:controltower:us-east-2::control/RYZTTCAMHLSE  
• **US West (Oregon)**  
arn:aws:controltower:us-west-2::control/NNKPEWE0GGGE  
• **Canada (Central)**  
arn:aws:controltower:ca-central-1::control/MWRRAOTEOGZF  
• **Asia Pacific (Sydney)**  
arn:aws:controltower:ap-southeast-2::control/OEWTPHSMPPGY  
• **Asia Pacific (Singapore)**  
arn:aws:controltower:ap-southeast-1::control/GKEKTSNJM0ZE  
• **Europe (Frankfurt)**  
arn:aws:controltower:eucentral-1::control/CGPVJAMLCPND  
• **Europe (Ireland)**  
arn:aws:controltower:eu-west-1::control/WYRBYRJBCFGJ  
• **Europe (London)**  
arn:aws:controltower:eu-west-2::control/JDVMZQVCXGKE  
• **Europe (Stockholm)**  
arn:aws:controltower:eu-north-1::control/UFOIHGEAASV  
• **Asia Pacific (Mumbai)**  
arn:aws:controltower:ap-south-1::control/KPCTMBEDXBVA |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.S3.11</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SH.S3.11**

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.S3.11</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.S3.11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/DCZQYLXVSENF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/VPORJKPKMJPX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/JGYTWTCQJI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/JQLFNEUIAMQV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/BOYXQVZJI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/RFTVUXSDPJB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/LDPZBYDVREOG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/UWDNCNQHITXR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/IFMFFIFRXKSP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/GMKHBZANUDII</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ZOLMKQEAZFIO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.S3.12           | • NIST 800-53 Rev 5 AC-2(1)  
• NIST 800-53 Rev 5 AC-3  
• NIST 800-53 Rev 5 AC-3(15)  
• NIST 800-53 Rev 5 AC-3(7)  
• NIST 800-53 Rev 5 AC-6  
• PCI DSS version 3.2.1  
• PCI DSS version 3.2.1 7.1.1  
• PCI DSS version 3.2.1 7.2.3 | • Enforce least privilege | • US East (N. Virginia)  
ar:n:aws:controltower:us-east-1::control/BKLSQSHMCKWE  
• US East (Ohio)  
ar:n:aws:controltower:us-east-2::control/KTPCIMELTXVN  
• US West (Oregon)  
ar:n:aws:controltower:us-west-2::control/SDRCGUWECWP  
• Canada (Central)  
ar:n:aws:controltower:ca-central-1::control/UNGOCLMLKNAH  
• Asia Pacific (Sydney)  
ar:n:aws:controltower:ap-southeast-2::control/IYZUNEUFKEBY  
• Asia Pacific (Singapore)  
ar:n:aws:controltower:ap-southeast-1::control/LEZPLYTTTVHV  
• Europe (Frankfurt)  
ar:n:aws:controltower:eu-central-1::control/CHIDBJOCXIKM  
• Europe (Ireland)  
ar:n:aws:controltower:eu-west-1::control/OCMEJBXKCSWF  
• Europe (London)  
ar:n:aws:controltower:eu-west-2::control/WQICVAPWAYEZ  
• Europe (Stockholm)  
ar:n:aws:controltower:eu-north-1::control/WYYTZOHUUUUA  
• Asia Pacific (Mumbai)  
ar:n:aws:controltower:ap-south-1::control/QYSZQBRLIVHX |
### SH.S3.13

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SH.S3.13</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 AU-9(2)</td>
<td></td>
<td>• Optimize costs</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ UDFSBDLMCKHG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-10</td>
<td></td>
<td>• Improve availability</td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ DDDWXYAMKSZD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-6(2)</td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ URNEOQLNCGXG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 CP-9</td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/ KPNAK0ZEARRB</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-5(2)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ TWWHDSTFPQWH</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SI-13(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.5.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 10.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 3.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/GCYHDAOAZEZQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/GYGNQCLCYPEX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/QPEXOFWNQEZY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/TWVXCPMNSEVC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/LIVWOXMJIWLK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/EKXYSHHNTFGF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/MNDGHSQGIHSP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/IIUDTDTJVTRUB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/HVDXLVVRSLWK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/XZDASAKGUPOCB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/BZWMUSQKKNPY</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.S3.2           | • CIS AWS Benchmark 1.4 3.3  
• NIST 800-53 Rev 5 AC-21  
• NIST 800-53 Rev 5 AC-3  
• NIST 800-53 Rev 5 AC-3(7)  
• NIST 800-53 Rev 5 AC-4  
• NIST 800-53 Rev 5 AC-4(21)  
• NIST 800-53 Rev 5 AC-6  
• NIST 800-53 Rev 5 SC-7  
• NIST 800-53 Rev 5 SC-7(11)  
• NIST 800-53 Rev 5 SC-7(16)  
• NIST 800-53 Rev 5 SC-7(20)  
• NIST 800-53 Rev 5 SC-7(21)  
• NIST 800-53 Rev 5 SC-7(3)  
• NIST 800-53 Rev 5 SC-7(4)  
• NIST 800-53 Rev 5 SC-7(9)  
• PCI DSS version 3.2.1 1.2.1  
• PCI DSS version 3.2.1 1.3  
• PCI DSS version 3.2.1 1.3.1  
• PCI DSS version 3.2.1 1.3.2  
• PCI DSS version 3.2.1 1.3.4  
• PCI DSS version 3.2.1 1.3.6  
• PCI DSS version 3.2.1 2.2 | • Enforce least privilege | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/ EUQZPHTPYCGS  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/ UWARQHCUKTWT  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/ KCEBTQBTIPXWJ  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/ BPFIVHGTXOGG  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/ WUHZWRYEJOUJ  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/ LBKNUOHEWDPK  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/ UXNOYWDKLNL  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/ RNRFWUNSMVZ0  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/ TNVEODWBLQQT  
• Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/ OELOVGEWUICF  
• Asia Pacific (Mumbai)  
arn:aws:controltower:ap-south-1::control/ WQFTUQVOYPUO |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.S3.3</td>
<td></td>
<td>Enforce least privilege</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/OXQICLGDWVUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/NRDDEBVYUNQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/TSWBJXBIHDP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/AQQIXVCNLDYX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/YJQGDKELZCKA</td>
</tr>
</tbody>
</table>

- **PCI DSS version 3.2.1**
  - **Asia Pacific (Seoul)** arn:aws:controltower:ap-northeast-2::control/JVTGANIDGRYD
  - **Asia Pacific (Tokyo)** arn:aws:controltower:ap-northeast-1::control/FNAGBSJDNSMR
  - **Europe (Paris)** arn:aws:controltower:eu-west-3::control/DNOPTVORREC
  - **South America (São Paulo)** arn:aws:controltower:sa-east-1::control/LEZLNXWHNOBS
  - **US West (N. California)** arn:aws:controltower:us-west-1::control/EUHTMNOVXIUP

- **CIS AWS Benchmark 1.4 3.3**
  - **US East (N. Virginia)** arn:aws:controltower:us-east-1::control/OXQICLGDWVUS
  - **US East (Ohio)** arn:aws:controltower:us-east-2::control/NRDDEBVYUNQ
  - **US West (Oregon)** arn:aws:controltower:us-west-2::control/TSWBJXBIHDP
  - **Canada (Central)** arn:aws:controltower:ca-central-1::control/AQQIXVCNLDYX
  - **Asia Pacific (Sydney)** arn:aws:controltower:ap-southeast-2::control/YJQGDKELZCKA

- **NIST 800-53 Rev 5 AC-21**
  - **NIST 800-53 Rev 5 AC-3**
  - **NIST 800-53 Rev 5 AC-3(7)**
  - **NIST 800-53 Rev 5 AC-4**
  - **NIST 800-53 Rev 5 AC-4(21)**
  - **NIST 800-53 Rev 5 AC-6**
  - **NIST 800-53 Rev 5 SC-7**
  - **NIST 800-53 Rev 5 SC-7(11)**
  - **NIST 800-53 Rev 5 SC-7(16)**
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) &lt;br&gt;arn:aws:controltower:ap-southeast-1::control/LEHHNNZDYPSG</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) &lt;br&gt;arn:aws:controltower:eu-central-1::control/ZTSEFWMWFBI</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>• Europe (Ireland) &lt;br&gt;arn:aws:controltower:eu-west-1::control/DWNOGUJYEE</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>• Europe (London) &lt;br&gt;arn:aws:controltower:eu-west-2::control/ZQLF5HMBNQF</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) &lt;br&gt;arn:aws:controltower:eu-north-1::control/LTIVGDURRHH</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) &lt;br&gt;arn:aws:controltower:ap-south-1::control/MUMHADVUHEXE</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) &lt;br&gt;arn:aws:controltower:ap-northeast-2::control/UQF5BKMVRVQK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) &lt;br&gt;arn:aws:controltower:ap-northeast-1::control/IOXPTFPAGNK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>• Europe (Paris) &lt;br&gt;arn:aws:controltower:eu-west-3::control/NCGLTHGVRQUZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>• South America (São Paulo) &lt;br&gt;arn:aws:controltower:sa-east-1::control/NQYKUEVXJMTQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>• US West (N. California) &lt;br&gt;arn:aws:controltower:us-west-1::control/LQHXMREDMHB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>•</td>
</tr>
</tbody>
</table>
### SH.S3.4

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.S3.4</td>
<td>• CIS AWS Benchmark 1.4 2.1.1</td>
<td>• Encrypt data at rest</td>
<td>• <strong>US East (N. Virginia)</strong> arn:aws:controltower:us-east-1::control/QQJFQYFRBAWX</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AU-9</td>
<td></td>
<td>• <strong>US East (Ohio)</strong> arn:aws:controltower:us-east-2::control/VALKTBQGPLPC</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CA-9(1)</td>
<td></td>
<td>• <strong>US West (Oregon)</strong> arn:aws:controltower:us-west-2::control/DUHWVFQKWRLV</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 CM-3(6)</td>
<td></td>
<td>• <strong>Canada (Central)</strong> arn:aws:controltower:ca-central-1::control/KQHOHKCWINNM</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-13</td>
<td></td>
<td>• <strong>Asia Pacific (Sydney)</strong> arn:aws:controltower:ap-southeast-2::control/VAAXOMVQGSPM</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28</td>
<td></td>
<td>• <strong>Asia Pacific (Singapore)</strong> arn:aws:controltower:ap-southeast-1::control/TUEQ00DYXPVD</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-28(1)</td>
<td></td>
<td>• <strong>Europe (Frankfurt)</strong> arn:aws:controltower:eu-central-1::control/WQQFDWRXNWUU</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(10)</td>
<td></td>
<td>• <strong>Europe (Ireland)</strong> arn:aws:controltower:eu-west-1::control/BSUTLBPCTXDCX</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 SI-7(6)</td>
<td></td>
<td>• <strong>Europe (London)</strong> arn:aws:controltower:eu-west-2::control/IZAMVIVLXXXI</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.5</td>
<td></td>
<td>• <strong>Europe (Stockholm)</strong> arn:aws:controltower:eu-north-1::control/LNRHAOUQAFJW</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 10.5.2</td>
<td></td>
<td>• <strong>Asia Pacific (Mumbai)</strong> arn:aws:controltower:ap-south-1::control/NUPCQDXDAERA</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2398
### SH.S3.5

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Encrypt data in transit</td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/KNNNOLOAVMAG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ZCJUOGINWRJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/APGXGWZTZKSMW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sa-east-1::control/DJILLKIIGNGY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/OPYDRCPOJSJTKL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.S3.5</td>
<td>CIS AWS Benchmark 1.4 2.1.2</td>
<td>Encrypt data in transit</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/QPIQHVFHRWEY</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-17(2)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/PZTPBSALVUYK</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/HYHNWCTPBF</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 IA-5(1)</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/RQKKRCWIIKY</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-12(3)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/MYWMNVQHDBPC</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
</tr>
<tr>
<td>SC-8(1)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/BIELWGHJKGPA</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt)</td>
</tr>
<tr>
<td>SC-8(2)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/AAHCKLBMUEV</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5</td>
<td></td>
<td></td>
<td>• Europe (Ireland)</td>
</tr>
<tr>
<td>SI-7(6)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/VPQXZTSPWVVX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>• Europe (London)</td>
</tr>
<tr>
<td>2.2</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/PRNYXIYGVXN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>• Europe (Stockholm)</td>
</tr>
<tr>
<td>4.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/VDLZMBUFATGB</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
</tr>
<tr>
<td>8.2.1</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/OJXAXIGHYFQW</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/OVTXWKOBZPZR</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/ULMDATXUHGD</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/UZYVCOLSTZUL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/PNULPQUNPXRO</td>
</tr>
</tbody>
</table>
## SH.S3.6

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.S3.6            | • NIST 800-53 Rev 5 CA-9(1)  
• NIST 800-53 Rev 5 CM-2  
• PCI DSS version 3.2.1 7.1.1  
• PCI DSS version 3.2.1 7.2.3 | • Enforce least privilege | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/KTMNKSWVOVXN  
• US East (Ohio) arn:aws:controltower:us-east-2::control/FZXEO1LVPPYH  
• US West (Oregon) arn:aws:controltower:us-west-2::control/TIXQXDCZZIOI  
• Canada (Central) arn:aws:controltower:ca-central-1::control/TSCSMHZBPA0D  
• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/FGNEAZJPHZUU  
• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/AZGSO5ZUWIKH  
• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/QNJWDZCIVDCH  
• Europe (Ireland) arn:aws:controltower:eu-west-1::control/XMwBZXRYGTMC  
• Europe (London) arn:aws:controltower:eu-west-2::control/MGObAFXHMMUH  
• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/QKUSDPBWBMZG  
• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/DBISINHULBGV |
### SH.S3.8

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.S3.8            | CIS AWS Benchmark 1.4 2.1.5 | Enforce least privilege | • US East (N. Virginia) arn:aws:controltower:us-east-1::control/ BPOQJKFITSWH  
• US East (Ohio) arn:aws:controltower:us-east-2::control/ ZAAZUGROKPH  
• US West (Oregon) arn:aws:controltower:us-west-2::control/ PDKYAANJEWJE  
• Canada (Central) arn:aws:controltower:ca-central-1::control/ RXZMQUEO1MGW  
• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ HISI0KDAAMQR |
<p>|                    | NIST 800-53 Rev 5 AC-21 |                                | • Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ QJAYWXCXHDN |
|                    | NIST 800-53 Rev 5 AC-3  |                                | • Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ XUHLFSNTEWFP |
|                    | NIST 800-53 Rev 5 AC-3(7) |                                | • Europe (Paris) arn:aws:controltower:europe-west-3::control/ AARUNPWJDELC |
|                    | NIST 800-53 Rev 5 AC-4  |                                | • South America (São Paulo) arn:aws:controltower:sao-east-1::control/ WKWUMNSAAANN |
|                    | NIST 800-53 Rev 5 SC-7 |                                |                                 |
|                    | NIST 800-53 Rev 5 SC-7(11) |                            |                                 |
|                    | NIST 800-53 Rev 5 SC-7(16) |                            |                                 |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(20)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/NAQFFFPSERVD</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/OGUOLZTXJTVW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/SYIHHFFPXEMOL</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/LMEJYAPAVGXW</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/YUTTRNVCQCIJ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/HTLH5BYDSNKX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/LNGHEQDYEQLV</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/LUJLLEORWUBI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/PNWCPAETGOAL</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:saeast-1::control/YFMJMXPVFUKX</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/RTJHNMHBMMAJ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| SH.S3.9            | • CIS AWS Benchmark 1.4 3.6  
• NIST 800-53 Rev 5 AC-2(4)  
• NIST 800-53 Rev 5 AC-4(26)  
• NIST 800-53 Rev 5 AC-6(9)  
• NIST 800-53 Rev 5 AU-10  
• NIST 800-53 Rev 5 AU-12  
• NIST 800-53 Rev 5 AU-2  
• NIST 800-53 Rev 5 AU-3  
• NIST 800-53 Rev 5 AU-6(3)  
• NIST 800-53 Rev 5 AU-6(4)  
• NIST 800-53 Rev 5 CA-7  
• NIST 800-53 Rev 5 SC-7(9)  
• NIST 800-53 Rev 5 SI-3(8)  
• NIST 800-53 Rev 5 SI-4(20)  
• NIST 800-53 Rev 5 SI-7(8)  
• PCI DSS version 3.2.1 10.1  
• PCI DSS version 3.2.1 10.2.1  
• PCI DSS version 3.2.1 10.2.3  
• PCI DSS version 3.2.1 10.2.4  
• PCI DSS version 3.2.1 10.3.1  
• PCI DSS version 3.2.1 10.3.2  
• PCI DSS version 3.2.1 10.3.3 | • Establish logging and monitoring | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/QMwZZISJNvG  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/XEYTESFJVRU  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/UMKAXUFOAXGS  
• Canada (Central)  
arn:aws:controltower:ca-central-1::control/GFOPVBQVUFV  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/ZPOEOPATKJIC  
• Asia Pacific (Singapore)  
arn:aws:controltower:ap-southeast-1::control/MXLTHQZBCTDQ  
• Europe (Frankfurt)  
arn:aws:controltower:eu-central-1::control/YWKWSZSNPSI  
• Europe (Ireland)  
arn:aws:controltower:eu-west-1::control/HVKIHTICWHEL  
• Europe (London)  
arn:aws:controltower:eu-west-2::control/TZQ1UGSNXXNG  
• Europe (Stockholm)  
arn:aws:controltower:eu-north-1::control/FSDRKINNPBD  
• Asia Pacific (Mumbai)  
arn:aws:controltower:ap-south-1::control/THYMKLHAVMWC |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.SNS.1</td>
<td>PCI DSS version 3.2.1 10.3.4, PCI DSS version 3.2.1 10.3.5, PCI DSS version 3.2.1 10.3.6, PCI DSS version 3.2.1 2.2</td>
<td>Encrypt data at rest</td>
<td>Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/XKQRAYJCRJRS, Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/KHOWWKTFTVWNS, Europe (Paris) arn:aws:controltower:eu-west-3::control/XEVOZIAAWYKS, South America (São Paulo) arn:aws:controltower:sa-east-1::control/YZMEODWAHVDL, US West (N. California) arn:aws:controltower:us-west-1::control/ZJRFWTSWEVB</td>
</tr>
</tbody>
</table>

### SH.SNS.1

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective, by Region</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------------------</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/EXVFJOMQYEPZ</td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/VFLOWQDNMCQF</td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/ATAMJPTZTAR</td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/AQEPYMTMFBVZ</td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/FZCVBXYXZEPW</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/WDDSEPQGQGQL</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/MJERXRWAXBXQL</td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/FQDFREMVXBDY</td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/MPWYPMECHJHD</td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/YYTATHJHTJHD</td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/EIHPUZHBQTSB</td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
</tbody>
</table>
| SH.SNS.2           | • NIST 800-53 Rev 5 AU-12  
|                    | • NIST 800-53 Rev 5 AU-2  
|                    | • PCI DSS version 3.2.1 10.1 | • Establish logging and monitoring | • **US East (N. Virginia)**  
|                    |                                      |                                 | arn:aws:controltower:us-east-1::control/UJGAATDFQPEB |
|                    |                                      |                                 | • **US East (Ohio)**  
|                    |                                      |                                 | arn:aws:controltower:us-east-2::control/OZGKTTQNVJH |
|                    |                                      |                                 | • **US West (Oregon)**  
|                    |                                      |                                 | arn:aws:controltower:us-west-2::control/BQNJQZDDAPMH |
|                    |                                      |                                 | • **Canada (Central)**  
|                    |                                      |                                 | arn:aws:controltower:ca-central-1::control/ZASOMECURBPO |
|                    |                                      |                                 | • **Asia Pacific (Sydney)**  
|                    |                                      |                                 | arn:aws:controltower:ap-southeast-2::control/YVTYTAMYHBIH |
|                    |                                      |                                 | • **Asia Pacific (Singapore)**  
|                    |                                      |                                 | arn:aws:controltower:ap-southeast-1::control/SPUXXPIXTYXG |
|                    |                                      |                                 | • **Europe (Frankfurt)**  
|                    |                                      |                                 | arn:aws:controltower:eu-central-1::control/FMBNUPQTJINIA |
|                    |                                      |                                 | • **Europe (Ireland)**  
|                    |                                      |                                 | arn:aws:controltower:eu-west-1::control/PEPGKWQCCMZ |
|                    |                                      |                                 | • **Europe (London)**  
|                    |                                      |                                 | arn:aws:controltower:eu-west-2::control/IFFLSEGZZOUB |
|                    |                                      |                                 | • **Europe (Stockholm)**  
|                    |                                      |                                 | arn:aws:controltower:eu-north-1::control/CTOVFQDVJCJAA |
|                    |                                      |                                 | • **Asia Pacific (Mumbai)**  
|                    |                                      |                                 | arn:aws:controltower:ap-south-1::control/EFGSSRJSCQHJ |

2407
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.SQS.1           |           | Encrypt data at rest | • US East (N. Virginia)  
arn:aws:controltower:us-east-1::control/  
AQPZJAICUMOX  
• US East (Ohio)  
arn:aws:controltower:us-east-2::control/  
FSJSXWXDXTQG  
• US West (Oregon)  
arn:aws:controltower:us-west-2::control/  
DVIY0VZQBBHZ  
• Canada (Central)  
ar:n:aws:controltower:ca-central-1::control/  
YNRXRVWR0TER  
• Asia Pacific (Sydney)  
arn:aws:controltower:ap-southeast-2::control/  
ENSHCNPLXZHX |
|                    |           |                  | • Asia Pacific (Seoul)  
arn:aws:controltower:ap-northeast-2::control/  
EWMPINBPPCPC  
• Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-northeast-1::control/  
DPMVGVDIISI  
• Europe (Paris)  
ar:n:aws:controltower:eu-west-3::control/  
SODUCRSZJCQJ  
• South America (São Paulo)  
ar:n:aws:controltower:sa-east-1::control/  
MFCJPTYVEGUF  
• US West (N. California)  
ar:n:aws:controltower:us-west-1::control/  
NVTOKRXVIZCW |
|                    |           |                  | • South America (São Paulo)  
ar:n:aws:controltower:sa-east-1::control/  
MFCJPTYVEGUF  
• US West (N. California)  
ar:n:aws:controltower:us-west-1::control/  
NVTOKRXVIZCW |

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.SQS.1           |           |                  | • Asia Pacific (Seoul)  
ar:n:aws:controltower:ap-northeast-2::control/  
EWMPINBPPCPC  
• Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-northeast-1::control/  
DPMVGVDIISI  
• Europe (Paris)  
ar:n:aws:controltower:eu-west-3::control/  
SODUCRSZJCQJ  
• South America (São Paulo)  
ar:n:aws:controltower:sa-east-1::control/  
MFCJPTYVEGUF  
• US West (N. California)  
ar:n:aws:controltower:us-west-1::control/  
NVTOKRXVIZCW |
|                    |           |                  | • Asia Pacific (Seoul)  
ar:n:aws:controltower:ap-northeast-2::control/  
EWMPINBPPCPC  
• Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-northeast-1::control/  
DPMVGVDIISI  
• Europe (Paris)  
ar:n:aws:controltower:eu-west-3::control/  
SODUCRSZJCQJ  
• South America (São Paulo)  
ar:n:aws:controltower:sa-east-1::control/  
MFCJPTYVEGUF  
• US West (N. California)  
ar:n:aws:controltower:us-west-1::control/  
NVTOKRXVIZCW |

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.SQS.1           |           |                  | • Asia Pacific (Seoul)  
ar:n:aws:controltower:ap-northeast-2::control/  
EWMPINBPPCPC  
• Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-northeast-1::control/  
DPMVGVDIISI  
• Europe (Paris)  
ar:n:aws:controltower:eu-west-3::control/  
SODUCRSZJCQJ  
• South America (São Paulo)  
ar:n:aws:controltower:sa-east-1::control/  
MFCJPTYVEGUF  
• US West (N. California)  
ar:n:aws:controltower:us-west-1::control/  
NVTOKRXVIZCW |
|                    |           |                  | • Asia Pacific (Seoul)  
ar:n:aws:controltower:ap-northeast-2::control/  
EWMPINBPPCPC  
• Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-northeast-1::control/  
DPMVGVDIISI  
• Europe (Paris)  
ar:n:aws:controltower:eu-west-3::control/  
SODUCRSZJCQJ  
• South America (São Paulo)  
ar:n:aws:controltower:sa-east-1::control/  
MFCJPTYVEGUF  
• US West (N. California)  
ar:n:aws:controltower:us-west-1::control/  
NVTOKRXVIZCW |

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.SQS.1           |           |                  | • Asia Pacific (Seoul)  
ar:n:aws:controltower:ap-northeast-2::control/  
EWMPINBPPCPC  
• Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-northeast-1::control/  
DPMVGVDIISI  
• Europe (Paris)  
ar:n:aws:controltower:eu-west-3::control/  
SODUCRSZJCQJ  
• South America (São Paulo)  
ar:n:aws:controltower:sa-east-1::control/  
MFCJPTYVEGUF  
• US West (N. California)  
ar:n:aws:controltower:us-west-1::control/  
NVTOKRXVIZCW |
|                    |           |                  | • Asia Pacific (Seoul)  
ar:n:aws:controltower:ap-northeast-2::control/  
EWMPINBPPCPC  
• Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-northeast-1::control/  
DPMVGVDIISI  
• Europe (Paris)  
ar:n:aws:controltower:eu-west-3::control/  
SODUCRSZJCQJ  
• South America (São Paulo)  
ar:n:aws:controltower:sa-east-1::control/  
MFCJPTYVEGUF  
• US West (N. California)  
ar:n:aws:controltower:us-west-1::control/  
NVTOKRXVIZCW |

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
</table>
| SH.SQS.1           |           |                  | • Asia Pacific (Seoul)  
ar:n:aws:controltower:ap-northeast-2::control/  
EWMPINBPPCPC  
• Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-northeast-1::control/  
DPMVGVDIISI  
• Europe (Paris)  
ar:n:aws:controltower:eu-west-3::control/  
SODUCRSZJCQJ  
• South America (São Paulo)  
ar:n:aws:controltower:sa-east-1::control/  
MFCJPTYVEGUF  
• US West (N. California)  
ar:n:aws:controltower:us-west-1::control/  
NVTOKRXVIZCW |
|                    |           |                  | • Asia Pacific (Seoul)  
ar:n:aws:controltower:ap-northeast-2::control/  
EWMPINBPPCPC  
• Asia Pacific (Tokyo)  
ar:n:aws:controltower:ap-northeast-1::control/  
DPMVGVDIISI  
• Europe (Paris)  
ar:n:aws:controltower:eu-west-3::control/  
SODUCRSZJCQJ  
• South America (São Paulo)  
ar:n:aws:controltower:sa-east-1::control/  
MFCJPTYVEGUF  
• US West (N. California)  
ar:n:aws:controltower:us-west-1::control/  
NVTOKRXVIZCW |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/ NZDJTQADTXGS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/ SDQYTAUTUONG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/ KKB0VJ3ZKRXH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/ AREAYHKBSDVF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/ KBKQAWGKVVBK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/ OCRDEAEBDMNJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/ EAUIYIPCCXXZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/ ZDMVZWYXPYNG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/ EMOPJHNXCSWH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/ XCTLDMBSISCMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/ NKGRFJUHQFGJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>SH.SSM.1</td>
<td>• NIST 800-53 Rev 5 CA-9(1) &lt;br&gt; • NIST 800-53 Rev 5 CM-2 &lt;br&gt; • NIST 800-53 Rev 5 CM-2(2) &lt;br&gt; • NIST 800-53 Rev 5 CM-8 &lt;br&gt; • NIST 800-53 Rev 5 CM-8(1) &lt;br&gt; • NIST 800-53 Rev 5 CM-8(2) &lt;br&gt; • NIST 800-53 Rev 5 CM-8(3) &lt;br&gt; • NIST 800-53 Rev 5 SA-15(2) &lt;br&gt; • NIST 800-53 Rev 5 SA-15(8) &lt;br&gt; • NIST 800-53 Rev 5 SA-3 &lt;br&gt; • NIST 800-53 Rev 5 SI-2(3) &lt;br&gt; • PCI DSS version 3.2.1 2.4</td>
<td>• Manage vulnerabilities &lt;br&gt; • Protect configurations</td>
<td>• US East (N. Virginia) &lt;br&gt; arn:aws:controltower:us-east-1::control/LYPKGNLPNWHL &lt;br&gt; • US East (Ohio) &lt;br&gt; arn:aws:controltower:us-east-2::control/STNJGGCQJCIW &lt;br&gt; • US West (Oregon) &lt;br&gt; arn:aws:aws:controltower:us-west-2::control/MJMHFBPPWLJA &lt;br&gt; • Canada (Central) &lt;br&gt; arn:aws:controltower:ca-central-1::control/XOUNWPBZJNVB &lt;br&gt; • Asia Pacific (Sydney) &lt;br&gt; arn:aws:controltower:ap-southeast-2::control/RLRUQAVPCA2I &lt;br&gt; • Asia Pacific (Singapore) &lt;br&gt; arn:aws:controltower:ap-southeast-1::control/NWSTAJGOQBJJ &lt;br&gt; • Europe (Frankfurt) &lt;br&gt; arn:aws:controltower:eucentral-1::control/RAPYUEVJEJBI &lt;br&gt; • Europe (Ireland) &lt;br&gt; arn:aws:controltower:europe-west-1::control/RXNSTMQLKTSN &lt;br&gt; • Europe (London) &lt;br&gt; arn:aws:controltower:europe-west-2::control/ZQSTFGETEWJA &lt;br&gt; • Europe (Stockholm) &lt;br&gt; arn:aws:controltower:europe-north-1::control/KSKCLFWRTEJP &lt;br&gt; • Asia Pacific (Mumbai) &lt;br&gt; arn:aws:controltower:ap-south-1::control/STITIQJTN6SD</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>SH.SSM.2</td>
<td></td>
<td>Manage vulnerabilities</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/RDSQHZJNNJYD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/JSMHJHLY0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/FFOYUQFGQHM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/NMLXRBHLMGSL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/LARTAAZXHQGH</td>
</tr>
</tbody>
</table>

| SH.SSM.2           |           |                  | • Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/SOADYNVMUUSE |
|                    |           |                  | • Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/HGKHYMQCWXZ |
|                    |           |                  | • Europe (Paris) arn:aws:controltower:eu-west-3::control/QHTLZMCZKVAA |
|                    |           |                  | • South America (São Paulo) arn:aws:controltower:sa-east-1::control/PWIULRITCIK |
|                    |           |                  | • US West (N. California) arn:aws:controltower:us-west-1::control/AHGSLJBGRQD |

<p>| SH.SSM.2           |           |                  | • NIST 800-53 Rev 5 CM-8(3) |
|                    |           |                  | • NIST 800-53 Rev 5 SI-2 |
|                    |           |                  | • NIST 800-53 Rev 5 SI-2(2) |
|                    |           |                  | • NIST 800-53 Rev 5 SI-2(3) |
|                    |           |                  | • NIST 800-53 Rev 5 SI-2(4) |
|                    |           |                  | • NIST 800-53 Rev 5 SI-2(5) |
|                    |           |                  | • PCI DSS version 3.2.1 2.2 |
|                    |           |                  | • PCI DSS version 3.2.1 6.2 |</p>
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific (Singapore)</td>
<td></td>
<td>• Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/ZAEIEMPTKSGD</td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td></td>
<td>• Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/HSENLNFBHIHTF</td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td></td>
<td>• Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/KUSFIFXCWYPN</td>
</tr>
<tr>
<td>Europe (London)</td>
<td></td>
<td>• Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/OMNUMQWSKXCR</td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td></td>
<td>• Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/UAFFTPQZIXDJ</td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td></td>
<td>• Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/CWXNZUPFQUJG</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/YPZMNGTROKXV</td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/GYGBKSAI0IXS</td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td></td>
<td>• Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/NZGSDZDQPQIL</td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td></td>
<td>• South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/LNHSBYVVRSTVG</td>
</tr>
<tr>
<td>US West (N. California)</td>
<td></td>
<td>• US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/OCZSGXLBRYOH</td>
</tr>
</tbody>
</table>
### SH.SSM.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.SSM.3</td>
<td>NIST 800-53 Rev 5 CA-9(1)</td>
<td>Manage vulnerabilities</td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/KBRZYOMGLHXE</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2</td>
<td>Protect configurations</td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/FUUFUNICZWAN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-2(2)</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/RDXQFIOFSBGG</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-8</td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/FVQFKOJUNLOY</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-8(1)</td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/SFESCRTZJFXB</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 CM-8(3)</td>
<td></td>
<td>Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/DWXVJLGLHICA</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SI-2(3)</td>
<td></td>
<td>Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/BUFFWCQVUNMG</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>Europe (Ireland) arn:aws:controltower:eu-west-1::control/EQMJTEGXWFOU</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 6.2</td>
<td></td>
<td>Europe (London) arn:aws:controltower:eu-west-2::control/ICKTKXTHFZDK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Europe (Stockholm) arn:aws:controltower:eu-north-1::control/SLNLDXHNYMTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/CVEGWZBOLQVU</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.SSM.4</td>
<td></td>
<td>Limit network access</td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/JG5JATCRD0GPGX</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/VAYHIYPEEDS</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/VWFNFRROBVOX</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/VJYRNFSGCXM</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/SABQDVMDUTVI</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/XRDYUTEVJMMA</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/UDAQAMUOGDDV</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/IUFHQEAVVYUK</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/ATGIDTBHUTW</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:saeast-1::control/ZGSSXXGZVYKA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/MNCPCUXYPLYN</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3</td>
<td></td>
<td>US East (Ohio) Arn:aws:controltower:us-east-2::control/ TMAFWQWAQ0AY</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>US West (Oregon) Arn:aws:controltower:us-west-2::control/ KSWI0QETFUQN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-4</td>
<td></td>
<td>Canada (Central) Arn:aws:controltower:ca-central-1::control/ SQZAFTGIIXLN</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>Asia Pacific (Sydney) Arn:aws:controltower:ap-southeast-2::control/ LYBVPJQFKFUM</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7</td>
<td></td>
<td>Asia Pacific (Singapore) Arn:aws:controltower:ap-southeast-1::control/ OIWALMEQVBYZ</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(11)</td>
<td></td>
<td>Europe (Frankfurt) Arn:aws:controltower:eu-central-1::control/ WRTYZGZDWCOY</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(16)</td>
<td></td>
<td>Europe (Ireland) Arn:aws:controltower:eu-west-1::control/ NUSFXDTJ3WGYD</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td>Asia Pacific (Mumbai) Arn:aws:controltower:ap-south-1::control/ DGBAJFF0FTUB</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/ SNGBHAIQCTEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/ JICNYNFRVRQX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/ IPWMCVULQVXI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:south-america-east-1::control/ LXFEKJNOJXFJ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/ YMPZFEHCRVWM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.SageMaker.2</td>
<td></td>
<td>• Limit network access</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/ PMUACTTBWUIR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ VDBLYABRAGVK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/ EWVTVCSCSKQUM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:canada-central-1::control/ RAPZNYOHIVYB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/ XWKLTVYDAVXSW</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
<td></td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/TTNXTLKRLUQI</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(3)</td>
<td></td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/UNIXXNPDKKY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(4)</td>
<td></td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/UNXQJQANTY</td>
</tr>
<tr>
<td>• NIST 800-53 Rev 5 SC-7(9)</td>
<td></td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/CQECIKXXJUN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.2.1</td>
<td></td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/XJUIETBSJ0WP</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3</td>
<td></td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/PLPATENUQNNZ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.1</td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/O1KBJXEOZOYQ</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.2</td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/MCKMAWQKBWTG</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.4</td>
<td></td>
<td></td>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/QRSGKHVBVZGRN</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 1.3.6</td>
<td></td>
<td></td>
<td>• South America (São Paulo) arn:aws:controltower:sao-east-1::control/UUWNUMSGTIYY</td>
</tr>
<tr>
<td>• PCI DSS version 3.2.1 2.2.2</td>
<td></td>
<td></td>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/PLVBDYQNDLK</td>
</tr>
</tbody>
</table>
## SH.SageMaker.3

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.SageMaker.3</td>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td>• Enforce least privilege</td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/FLKLKRKSKRQQR</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/ZYWOLSUFYNHX</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(7)</td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/LWNZCRKRHRHB</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6</td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/GBJUIOMQLVOT</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6(10)</td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/GHYJJWOGIPDS</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-6(2)</td>
<td></td>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/HPLWOBUPSFSA</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 2.2</td>
<td></td>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/OJPMWQSLZIGR</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.1.1</td>
<td></td>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/QTVCCTBLIKXTQ</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.1.2</td>
<td></td>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/PHGBYKRRHVRZN</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.2.1</td>
<td></td>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/QOYZOFZYUVBP</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 7.2.2</td>
<td></td>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/LMYFYNKDCOKV</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 8.1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SH.SecretsManager.1</td>
<td></td>
<td>Manage secrets</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-2(1)</td>
<td></td>
<td>US East (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>US East (Ohio)</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 8.2.4</td>
<td></td>
<td>US West (Oregon)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney)</td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/GRZCUGFGEKEG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/HLEYUIBPLNQU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/YRYFWFSNQQEB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>South America</td>
<td>arn:aws:controltower:sa-east-1::control/VUPWZBFWTD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(São Paulo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/JBTRXDPZMK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US East (N. Virginia)</td>
<td>arn:aws:controltower:us-east-1::control/NVINDDWBCC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US East (Ohio)</td>
<td>arn:aws:controltower:us-east-2::control/YRBOGLMRVCT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>US West (Oregon)</td>
<td>arn:aws:controltower:us-west-2::control/FSFKVERCZUL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canada (Central)</td>
<td>arn:aws:controltower:ca-central-1::control/ABZHCĐTAXCTD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Sydney)</td>
<td>arn:aws:controltower:ap-southeast-2::control/PIPRUKNSOCX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/ABWGMGKAZWFBZ</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/VPLLFIYPDZ</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/PPXTTTZIQJWC</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/PAPTNHXGBYVU</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/CKUSCVHVNIR</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/VCDOMBQOZAKF</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/GPALVEMRIOYJ</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/LSEQYMIVIBNA</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/DZAFJWOHCOH</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/IHQTQRKAFFULV</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/NPFKTIKTONEU</td>
</tr>
</tbody>
</table>
### SH.SecretsManager.2

<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.SecretsManager.2</td>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td>• Manage secrets</td>
<td>• <strong>US East (N. Virginia)</strong> arn:aws:controltower:us-east-1::control/WMGPNBVGAVFK</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>• <strong>US East (Ohio)</strong> arn:aws:controltower:us-east-2::control/NBDBWRQFNMMA</td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.4</td>
<td></td>
<td>• <strong>US West (Oregon)</strong> arn:aws:controltower:us-west-2::control/MEFMYKVDMLQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Canada (Central)</strong> arn:aws:controltower:ca-central-1::control/STQSAQDRYTI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Asia Pacific (Sydney)</strong> arn:aws:controltower:ap-southeast-2::control/HOQMRBMNIKAW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Asia Pacific (Singapore)</strong> arn:aws:controltower:ap-southeast-1::control/ODPPJQBTCOTX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Frankfurt)</strong> arn:aws:controltower:eu-central-1::control/NNBWANKZRGIB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Ireland)</strong> arn:aws:controltower:eu-west-1::control/STHNSKDZTLYY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (London)</strong> arn:aws:controltower:eu-west-2::control/HKCSAJCXYOBE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• <strong>Europe (Stockholm)</strong> arn:aws:controltower:eu-north-1::control/ZASYFRCXDOII</td>
</tr>
</tbody>
</table>
|                    |                                                                          |                   | • **Asia Pacific (Mumbai)** arn:aws:controltower:ap-south-1::control/LOGBIWTPOLF}
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.SecretsManager.3</td>
<td></td>
<td>Manage secrets</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-2(1)</td>
<td></td>
<td>US East (N. Virginia) arn:aws:controltower:us-east-1::control/QURRRKYALIYF</td>
</tr>
<tr>
<td></td>
<td>NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
<td>US East (Ohio) arn:aws:controltower:us-east-2::control/DPKZCRDWWFPC</td>
</tr>
<tr>
<td></td>
<td>PCI DSS version 3.2.1 8.1.4</td>
<td></td>
<td>US West (Oregon) arn:aws:controltower:us-west-2::control/IVSZZVBQUNGM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Canada (Central) arn:aws:controltower:ca-central-1::control/ZKLBBEFBUMDR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/FNKXYQS2MWMQ</td>
</tr>
</tbody>
</table>

### SH.SecretsManager.3
- **NIST 800-53 Rev 5 AC-2(1)**
- **NIST 800-53 Rev 5 AC-3(15)**
- **PCI DSS version 3.2.1 8.1.4**
- **Manage secrets**
- **US East (N. Virginia)** arn:aws:controltower:us-east-1::control/QURRRKYALIYF
- **US East (Ohio)** arn:aws:controltower:us-east-2::control/DPKZCRDWWFPC
- **US West (Oregon)** arn:aws:controltower:us-west-2::control/IVSZZVBQUNGM
- **Canada (Central)** arn:aws:controltower:ca-central-1::control/ZKLBBEFBUMDR
- **Asia Pacific (Sydney)** arn:aws:controltower:ap-southeast-2::control/FNKXYQS2MWMQ
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Singapore) arn:aws:controltower:ap-southeast-1::control/DVRPVFHGPQI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Frankfurt) arn:aws:controltower:eu-central-1::control/PXRUQAOCPGNY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Ireland) arn:aws:controltower:eu-west-1::control/KRZOMDMLCLU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (London) arn:aws:controltower:eu-west-2::control/HNBTXEOY2HV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Stockholm) arn:aws:controltower:eu-north-1::control/GIFLRYEYHEHL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai) arn:aws:controltower:ap-south-1::control/DRWDDUMJXGDX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Seoul) arn:aws:controltower:ap-northeast-2::control/MSPQXYGOSXA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo) arn:aws:controltower:ap-northeast-1::control/FNOJORYKEHUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Europe (Paris) arn:aws:controltower:eu-west-3::control/GEOKFDUAJKJVU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• South America (São Paulo) arn:aws:controltower:saeast-1::control/SXONJYJKZCIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• US West (N. California) arn:aws:controltower:us-west-1::control/LXFYEIRGOXXV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>SH.SecretsManager.4</td>
<td>• NIST 800-53 Rev 5 AC-2(1)</td>
<td>• Manage secrets</td>
</tr>
<tr>
<td></td>
<td>• NIST 800-53 Rev 5 AC-3(15)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• PCI DSS version 3.2.1 8.2.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>SH.WAF.10</td>
<td></td>
<td>Limit network access</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>Asia Pacific (Singapore)</td>
<td>arn:aws:controltower:ap-southeast-1::control/VTALPZJYLJOC</td>
<td></td>
</tr>
<tr>
<td>Europe (Frankfurt)</td>
<td>arn:aws:controltower:eu-central-1::control/QNQKBYXYJBFJ</td>
<td></td>
</tr>
<tr>
<td>Europe (Ireland)</td>
<td>arn:aws:controltower:eu-west-1::control/XTRYNOHVVRZV</td>
<td></td>
</tr>
<tr>
<td>Europe (London)</td>
<td>arn:aws:controltower:eu-west-2::control/WWOBBVWIGZC</td>
<td></td>
</tr>
<tr>
<td>Europe (Stockholm)</td>
<td>arn:aws:controltower:eu-north-1::control/IZGQEGNHNXHB</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Mumbai)</td>
<td>arn:aws:controltower:ap-south-1::control/DZCJRBFWFYZA</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Seoul)</td>
<td>arn:aws:controltower:ap-northeast-2::control/SDFINEZEUSJY</td>
<td></td>
</tr>
<tr>
<td>Asia Pacific (Tokyo)</td>
<td>arn:aws:controltower:ap-northeast-1::control/QJDMOHVCOVCC</td>
<td></td>
</tr>
<tr>
<td>Europe (Paris)</td>
<td>arn:aws:controltower:eu-west-3::control/QHQLNZXRMJAS</td>
<td></td>
</tr>
<tr>
<td>South America (São Paulo)</td>
<td>arn:aws:controltower:sa-east-1::control/UXJNMZSDHBJG</td>
<td></td>
</tr>
<tr>
<td>US West (N. California)</td>
<td>arn:aws:controltower:us-west-1::control/KJPOXHOSICGE</td>
<td></td>
</tr>
</tbody>
</table>
## Control identifier | Framework | Control objective | Control API identifiers, by Region
--- | --- | --- | ---
SH.WAF.2 | • NIST 800-53 Rev 5 AC-4(21)  
• NIST 800-53 Rev 5 SC-7  
• NIST 800-53 Rev 5 SC-7(11)  
• NIST 800-53 Rev 5 SC-7(16)  
• NIST 800-53 Rev 5 SC-7(21)  
• PCI DSS version 3.2.1 6.6 | • Limit network access | • **US East (N. Virginia)**  
    arn:aws:controltower:us-east-1::control/HMAUVMDCZUKN  
• **US East (Ohio)**  
    arn:aws:controltower:us-east-2::control/IZERZZJXURES  
• **US West (Oregon)**  
    arn:aws:controltower:us-west-2::control/HVPCLRFIDJZZ  
• **Canada (Central)**  
    arn:aws:controltower:ca-central-1::control/PGGMESVIZFFP  
• **Asia Pacific (Sydney)**  
    arn:aws:controltower:ap-southeast-2::control/YJAUIVTFCYRJW  
• **Asia Pacific (Singapore)**  
    arn:aws:controltower:ap-southeast-1::control/MCGGLKIFMEWCCW  
• **Europe (Frankfurt)**  
    arn:aws:controltower:eu-central-1::control/YNDAPXATU10Y  
• **Europe (Ireland)**  
    arn:aws:controltower:eu-west-1::control/HOFUVZWNCTT  
• **Europe (London)**  
    arn:aws:controltower:eu-west-2::control/CBNPHNXPNMZG  
• **Europe (Stockholm)**  
    arn:aws:controltower:eu-north-1::control/ZYSHREMBYDPM  
• **Asia Pacific (Mumbai)**  
    arn:aws:controltower:ap-south-1::control/DLOLIYYVEKVV

2428
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH.WAF.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (N. Virginia) arn:aws:controltower:us-east-1::control/KONSORMMHUGB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US East (Ohio) arn:aws:controltower:us-east-2::control/KRWNFORBWPZ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (Oregon) arn:aws:controltower:us-west-2::control/LUZSILPCBBOK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Canada (Central) arn:aws:controltower:ca-central-1::control/TEFNOEPILSHB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Sydney) arn:aws:controltower:ap-southeast-2::control/CYGNEESCWXG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 AC-4(21)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(11)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(16)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• NIST 800-53 Rev 5 SC-7(21)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• PCI DSS version 3.2.1 6.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Limit network access</td>
<td></td>
</tr>
</tbody>
</table>

- **Asia Pacific (Seoul)** arn:aws:controltower:ap-northeast-2::control/TMCZBYVXXVTJL
- **Asia Pacific (Tokyo)** arn:aws:controltower:ap-northeast-1::control/DLQOSLPESPJ
- **Europe (Paris)** arn:aws:controltower:eu-west-3::control/TJGYMRUITIEJ
- **South America (São Paulo)** arn:aws:controltower:sa-east-1::control/SRGTMUGMZWSS
- **US West (N. California)** arn:aws:controltower:us-west-1::control/YGFAOMXOUWHM
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asia Pacific (Singapore)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-southeast-1::control/TFCNQSIYJFYC</td>
</tr>
<tr>
<td>• Europe (Frankfurt)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-central-1::control/VNFBLCXUEWUA</td>
</tr>
<tr>
<td>• Europe (Ireland)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-1::control/LSRTYDUWQTAE</td>
</tr>
<tr>
<td>• Europe (London)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-2::control/MUHOVSWIEMKQ</td>
</tr>
<tr>
<td>• Europe (Stockholm)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-north-1::control/HOLXVCBPWTJX</td>
</tr>
<tr>
<td>• Asia Pacific (Mumbai)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-south-1::control/BXSAQBULUHIU</td>
</tr>
<tr>
<td>• Asia Pacific (Seoul)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-2::control/YMBNBHLDYVOC</td>
</tr>
<tr>
<td>• Asia Pacific (Tokyo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-northeast-1::control/YRQA ZZKPDDPI</td>
</tr>
<tr>
<td>• Europe (Paris)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:eu-west-3::control/YPYYCPXNLRIH</td>
</tr>
<tr>
<td>• South America (São Paulo)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-east-1::control/MDUPXZXATPRTC</td>
</tr>
<tr>
<td>• US West (N. California)</td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-west-1::control/BUWKEMZLQXRX</td>
</tr>
<tr>
<td>Control identifier</td>
<td>Framework</td>
<td>Control objective</td>
<td>Control API identifiers, by Region</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------</td>
</tr>
</tbody>
</table>
| SH.WAF.4           | • NIST 800-53 Rev 5 CA-9(1)  
                        • NIST 800-53 Rev 5 CM-2  
                        • PCI DSS version 3.2.1 6.6 | • Limit network access | • **US East (N. Virginia)**  
                      arn:aws:controltower:us-east-1::control/EDXGDDVAPQUE  
                      • **US East (Ohio)**  
                      arn:aws:controltower:us-east-2::control/ZCXGWVXTGXMV  
                      • **US West (Oregon)**  
                      arn:aws:controltower:us-west-2::control/BFJEJSOYYREH  
                      • **Canada (Central)**  
                      arn:aws:controltower:ca-central-1::control/WQNMFDARNQD  
                      • **Asia Pacific (Sydney)**  
                      arn:aws:controltower:ap-southeast-2::control/LYLUSATWXWGZ  
                      • **Asia Pacific (Singapore)**  
                      arn:aws:controltower:ap-southeast-1::control/RWKJOQGECYVA  
                      • **Europe (Frankfurt)**  
                      arn:aws:controltower:eu-central-1::control/FXIUUDQLP00X  
                      • **Europe (Ireland)**  
                      arn:aws:controltower:eu-west-1::control/KGKFNHJFEKZX  
                      • **Europe (London)**  
                      arn:aws:controltower:eu-west-2::control/XKBUFFXZKSUP  
                      • **Europe (Stockholm)**  
                      arn:aws:controltower:eu-north-1::control/LKOWMWBVXXXZ  
                      • **Asia Pacific (Mumbai)**  
                      arn:aws:controltower:ap-south-1::control/ZGIHYMZBHXMV |
<table>
<thead>
<tr>
<th>Control identifier</th>
<th>Framework</th>
<th>Control objective</th>
<th>Control API identifiers, by Region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Seoul)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>northeast-2::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IVEPLVNEC5LY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Asia Pacific (Tokyo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:ap-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>northeast-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BEIGEZGMNWRL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Europe (Paris)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:euro</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>west-3::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GTNPVMNBKDAE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• South America (São Paulo)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:sa-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>east-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EVTTLKICADR2X</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• US West (N. California)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>arn:aws:controltower:us-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>west-1::control/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IZXJKPGKEK4KL</td>
</tr>
</tbody>
</table>
AWS glossary

For the latest AWS terminology, see the [AWS glossary](#) in the *AWS Glossary Reference*. 