# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>What Is AWS Control Tower?</td>
<td>1</td>
</tr>
<tr>
<td>Features</td>
<td>1</td>
</tr>
<tr>
<td>How AWS Control Tower interacts with other AWS services</td>
<td>2</td>
</tr>
<tr>
<td>Are You a First-Time User of AWS Control Tower?</td>
<td>2</td>
</tr>
<tr>
<td>How It Works</td>
<td>2</td>
</tr>
<tr>
<td>Structure of an AWS Control Tower Landing Zone</td>
<td>3</td>
</tr>
<tr>
<td>What happens when you set up a landing zone</td>
<td>3</td>
</tr>
<tr>
<td>What Are the Shared Accounts?</td>
<td>3</td>
</tr>
<tr>
<td>How Guardrails Work</td>
<td>9</td>
</tr>
<tr>
<td>How AWS Control Tower Works With StackSets</td>
<td>10</td>
</tr>
<tr>
<td>Plan your landing zone</td>
<td>10</td>
</tr>
<tr>
<td>Compare functionality</td>
<td>11</td>
</tr>
<tr>
<td>Launch AWS Control Tower in an Existing Organization</td>
<td>11</td>
</tr>
<tr>
<td>Launch AWS Control Tower in a New Organization</td>
<td>12</td>
</tr>
<tr>
<td>Terminology</td>
<td>13</td>
</tr>
<tr>
<td>Pricing</td>
<td>15</td>
</tr>
<tr>
<td>Setting up</td>
<td>15</td>
</tr>
<tr>
<td>Sign up for AWS</td>
<td>16</td>
</tr>
<tr>
<td>Create an IAM User</td>
<td>16</td>
</tr>
<tr>
<td>Set up MFA</td>
<td>18</td>
</tr>
<tr>
<td>Next Step</td>
<td>18</td>
</tr>
<tr>
<td>Getting started</td>
<td>19</td>
</tr>
<tr>
<td>Prerequisite: Automated pre-launch checks for your management account</td>
<td>19</td>
</tr>
<tr>
<td>Considerations for AWS Single Sign-On (AWS SSO) customers</td>
<td>19</td>
</tr>
<tr>
<td>Considerations for AWS Config and AWS CloudTrail customers</td>
<td>20</td>
</tr>
<tr>
<td>Requirements for your shared account email addresses</td>
<td>20</td>
</tr>
<tr>
<td>Expectations for landing zone configuration</td>
<td>21</td>
</tr>
<tr>
<td>Configure and launch your landing zone</td>
<td>21</td>
</tr>
<tr>
<td>Step 1. Review pricing and select your AWS Regions</td>
<td>22</td>
</tr>
<tr>
<td>Step 2. Configure your organizational units (OUs)</td>
<td>22</td>
</tr>
<tr>
<td>Step 3. Configure your shared accounts and encryption</td>
<td>22</td>
</tr>
<tr>
<td>Step 4. Review and set up the landing zone</td>
<td>24</td>
</tr>
<tr>
<td>Next steps</td>
<td>24</td>
</tr>
<tr>
<td>Limitations and quotas</td>
<td>26</td>
</tr>
<tr>
<td>Limitations in AWS Control Tower</td>
<td>26</td>
</tr>
<tr>
<td>Quotas for Integrated Services</td>
<td>26</td>
</tr>
<tr>
<td>Best practices for administrators</td>
<td>27</td>
</tr>
<tr>
<td>Explaining Access to Users</td>
<td>27</td>
</tr>
<tr>
<td>Explaining Resource Access</td>
<td>27</td>
</tr>
<tr>
<td>Explaining Preventive Guardrails</td>
<td>28</td>
</tr>
<tr>
<td>Administrative tips for landing zone setup</td>
<td>28</td>
</tr>
<tr>
<td>Recommendations for setting up groups, roles, and policies</td>
<td>29</td>
</tr>
<tr>
<td>Guidance for creating and modifying AWS Control Tower resources</td>
<td>29</td>
</tr>
<tr>
<td>Sign in as a Root User</td>
<td>30</td>
</tr>
<tr>
<td>AWS Organizations Guidance</td>
<td>31</td>
</tr>
<tr>
<td>AWS Single Sign-On Guidance</td>
<td>32</td>
</tr>
<tr>
<td>Account Factory Guidance</td>
<td>32</td>
</tr>
<tr>
<td>Guidance on Subscribing to SNS Topics</td>
<td>32</td>
</tr>
<tr>
<td>Guidance for KMS keys</td>
<td>32</td>
</tr>
<tr>
<td>AWS Control Tower and VPCs</td>
<td>33</td>
</tr>
<tr>
<td>CIDR and Peering for VPC and AWS Control Tower</td>
<td>34</td>
</tr>
<tr>
<td>Configuration update management</td>
<td>35</td>
</tr>
</tbody>
</table>
Enable guardrails ................................................................. 175
Concurrent detective guardrail deployment .............................................. 176
Optional guardrails ........................................................................... 176
Guardrails and compliance .................................................................. 176
  How can administrators review compliance? ........................................ 177
  AWS Control Tower guardrail compliance status ................................. 178
  Drift prevention and notification ...................................................... 179
  Guardrail compliance notifications by SNS ........................................ 180
Guardrail reference ........................................................................... 181
  Mandatory guardrails ...................................................................... 181
  Strongly recommended guardrails .................................................... 195
  Elective guardrails ......................................................................... 204
Integrated services ............................................................................ 226
  AWS CloudFormation ................................................................. 226
  CloudTrail .................................................................................. 226
  CloudWatch ............................................................................... 227
  AWS Config ............................................................................... 227
  IAM .......................................................................................... 227
  AWS Lambda ............................................................................. 227
  AWS Organizations ...................................................................... 227
  Considerations ............................................................................ 228
  Amazon S3 .................................................................................. 228
  AWS Service Catalog ................................................................. 228
  AWS SSO .................................................................................. 229
  Things to Know About SSO Accounts and AWS Control Tower ........ 230
  AWS SSO Groups for AWS Control Tower ..................................... 230
  Amazon SNS ............................................................................. 232
  Step Functions .......................................................................... 233
Security ............................................................................................. 234
  Data Protection .......................................................................... 234
  Encryption at Rest ....................................................................... 235
  Encryption in Transit ..................................................................... 235
  Restrict Access to Content .......................................................... 235
Identity and Access Management ......................................................... 235
  Authentication ............................................................................ 236
  Access Control .......................................................................... 237
  Overview of Managing Access ..................................................... 237
  Prevent confused deputy attacks .................................................. 240
  Using Identity-Based Policies (IAM Policies) .................................... 240
Compliance Validation ......................................................................... 245
Resilience ......................................................................................... 246
Infrastructure Security ........................................................................ 247
Logging and monitoring ...................................................................... 248
Monitoring ....................................................................................... 249
Logging AWS Control Tower Actions with AWS CloudTrail .................... 249
  AWS Control Tower Information in CloudTrail ............................... 249
  Example: AWS Control Tower Log File Entries .............................. 251
Lifecycle Events .................................................................................. 252
  CreateManagedAccount ............................................................. 254
  UpdateManagedAccount ............................................................ 254
  EnableGuardrail ......................................................................... 255
  DisableGuardrail ........................................................................ 256
  SetupLandingZone ...................................................................... 257
  UpdateLandingZone .................................................................... 258
  RegisterOrganizationalUnit ......................................................... 259
  DeregisterOrganizationalUnit ....................................................... 260
AWS Control Tower introduces Terraform account provisioning and customization 291
New lifecycle event available .......................................................................................... 291
AWS Control Tower enables nested OUs ........................................................................ 291
Detective guardrail concurrency ................................................................................... 292
Two new Regions available .......................................................................................... 293
Region deselection ........................................................................................................ 293
AWS Control Tower works with AWS Key Management Systems ................................ 295
Guardrails renamed, functionality unchanged ............................................................... 294
AWS Control Tower scans SCPs daily to check for drift ............................................... 294
Customized names for OUs and accounts ..................................................................... 294
AWS Control Tower landing zone version 2.7 ............................................................. 295
Three new AWS Regions available ............................................................................... 296
Govern selected Regions only ....................................................................................... 296
AWS Control Tower now extends governance to existing OUs in your AWS organizations .................................................. 296
AWS Control Tower provides bulk account updates ...................................................... 296
January - December 2020 ........................................................................................... 297
AWS Control Tower console now links to external AWS Config rules ......................... 297
AWS Control Tower now available in additional Regions ............................................... 298
Guardrail update .......................................................................................................... 298
AWS Control Tower console shows more detail about OUs and accounts ................... 298
Use AWS Control Tower to set up new multi-account AWS environments in AWS Organizations .................................................. 299
Customizations for AWS Control Tower solution .................................................... 299
General availability of AWS Control Tower version 2.3 ............................................ 299
Single-step account provisioning in AWS Control Tower .............................................. 300
AWS Control Tower decommissioning tool ................................................................. 300
AWS Control Tower lifecycle event notifications ....................................................... 301
January - December 2019 ........................................................................................... 301
General availability of AWS Control Tower version 2.2 ............................................ 301
New elective guardrails in AWS Control Tower .......................................................... 302
New detective guardrails in AWS Control Tower ......................................................... 302
AWS Control Tower accepts email addresses for shared accounts with different domains than the management account .................................................................................. 302
General availability of AWS Control Tower version 2.1 ............................................ 303
Document history ........................................................................................................ 304
AWS glossary ............................................................................................................... 309
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 Single Sign-on, 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 preventive and detective controls (guardrails). For example, you can use guardrails 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. 38).

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.

- **Guardrails** – A guardrail is a high-level rule that provides ongoing governance for your overall AWS environment. It's expressed in plain language. Two kinds of guardrails exist: preventive and detective. Three categories of guidance apply to the two kinds of guardrails: mandatory, strongly recommended, or elective. For more information about guardrails, see How Guardrails 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. 117).

- **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, guardrails enabled for policy enforcement, guardrails 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 Single Sign-On, and AWS Organizations. For more information, see Integrated services (p. 226).

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 guardrails 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 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. 10) and AWS multi-account strategy for your AWS Control Tower landing zone (p. 37).
2. If you’re ready to create your first landing zone, see Getting started with AWS Control Tower (p. 19).
3. For information on drift detection and prevention, see Detect and resolve drift in AWS Control Tower (p. 148).
4. For security details, see Security in AWS Control Tower (p. 234).
5. For information on updating your landing zone and member accounts, see Configuration update management in AWS Control Tower (p. 35).

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*. You can choose customized names for these shared accounts when you launch your landing zone. However, they cannot be renamed later.
- **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.
- **AWS SSO directory** – This directory houses your AWS SSO users. It defines the scope of permissions for each AWS SSO user.
- **AWS SSO 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 two shared accounts in the Security OU: the Log Archive account and the Audit account.
- Creates a cloud-native directory in AWS SSO, with preconfigured groups and single sign-on access.
- Applies 20 mandatory, preventive guardrails to enforce policies.
- Applies two mandatory, detective guardrails to detect configuration violations.
- Preventive guardrails are not applied to the management account.
- Except for the management account, guardrails 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. Deletions 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 (p. 29).

- When you enable optional guardrails (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 guardrails entering an unknown state. For more information, see Guardrail reference (p. 181).

What Are the Shared Accounts?

In AWS Control Tower, three shared accounts in your landing zone are provisioned automatically 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 guardrails.

**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>AWS CloudFormation</td>
<td>StackSets</td>
<td>AWSControlTowerBP-BASELINE-CLOUDTRAIL</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-CONFIG-MASTER (in version 2.6 and later)</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>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-WRITE-PROHIBITED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSControlTowerLoggingResources</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</td>
<td>Security Resources</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</td>
<td>Roles</td>
<td>AWSControlTowerAdmin</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td>AWSControlTowerStackSetRole</td>
</tr>
<tr>
<td>AWS Identity and Access</td>
<td>Policies</td>
<td>AWSControlTowerCloudTrailRolePolicy</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td>AWSControlTowerAdminPolicy</td>
</tr>
<tr>
<td>AWS Single Sign-On</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>AWS CloudTrail</td>
<td></td>
<td>AWSLogArchiveAdmins</td>
</tr>
<tr>
<td>AWS Identity and Access</td>
<td></td>
<td>AWSLogArchiveViewers</td>
</tr>
<tr>
<td>Management</td>
<td>Policies</td>
<td>AWSSecurityAuditors</td>
</tr>
<tr>
<td>AWS Single Sign-On</td>
<td></td>
<td>AWSSecurityAuditPowerUsers</td>
</tr>
<tr>
<td></td>
<td>Permission Sets</td>
<td>AWSServiceCatalogAdmins</td>
</tr>
<tr>
<td>AWS Identity and Access</td>
<td></td>
<td>AWSAdministratorAccess</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td>AWSPowerUserAccess</td>
</tr>
<tr>
<td>AWS Single Sign-On</td>
<td></td>
<td>AWSServiceCatalogAdminFullAccess</td>
</tr>
<tr>
<td></td>
<td>Permission Sets</td>
<td>AWSServiceCatalogEndUserAccess</td>
</tr>
<tr>
<td>AWS Identity and Access</td>
<td></td>
<td>AWSReadOnlyAccess</td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td>AWSOrganizationsFullAccess</td>
</tr>
</tbody>
</table>
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-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-ROLES-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StackSet-AWSControlTowerLoggingResources-</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>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/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>AWS service</td>
<td>Resource type</td>
<td>Resource Name</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------</td>
<td>---------------------------------------------------</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</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-SecurityNotifications</td>
</tr>
<tr>
<td>Service</td>
<td></td>
<td></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.
## 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>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>
<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>
</tbody>
</table>

**Resource types:**
- StackSet
- Aggregator
- AWS Config Rules
- Trail
- CloudWatch Event Rules
- CloudWatch Logs
### How Guardrails Work

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

Preventive guardrails prevent actions from occurring. For example, the elective guardrail called **Disallow Changes to Bucket Policy for Amazon S3 Buckets** (Previously called **Disallow Policy Changes to Log 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 guardrails detect specific events when they occur and log the action in CloudTrail. For example, the strongly recommended guardrail 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.
For those who are familiar with AWS: In AWS Control Tower preventive guardrails are implemented with Service Control Policies (SCPs). Detective guardrails are implemented with AWS Config rules.

Related Topics

- Guardrails in AWS Control Tower (p. 173)
- Detect and resolve drift in AWS Control Tower (p. 148)

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 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.

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. 37).

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. 11): This section is for customers who have existing AWS Organizations ready to bring into governance by AWS Control Tower.
- Launch AWS Control Tower in a New Organization (p. 12): This section is for customers without existing AWS Organizations, OUs, and accounts.
Note
If you already have an AWS Organizations landing zone, you can extend AWS Control Tower governance from the existing landing zone to some or all of your existing OUs and accounts within an organization. See Govern existing organizations and accounts.

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 guardrails 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. 277). Also, for information about moving accounts from one landing zone to another, see What if the account does not meet the prerequisites? (p. 109)

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 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 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 guardrails 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. 19).

For more information about how AWS Control Tower interacts with existing AWS Organizations, see Enable AWS Control Tower on organizations and accounts (p. 162).

**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. 16) document.

AWS Control Tower sets up an organization for you automatically when you don't have one set up.
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 guardrails 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.
- 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 guardrails. 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.
- 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 guardrail.
- 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 guardrails enabled on the OU apply to all accounts within it, including unenrolled ones. This is true because preventive guardrails 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 guardrails 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 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.

- **Guardrail**: A guardrail is a high-level rule that provides ongoing governance for your overall
  AWS Control Tower environment. Each guardrail enforces a single rule. Preventive guardrails are
  implemented with SCPs. Detective guardrails are implemented with AWS Config rules. For more
  information, see How Guardrails 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 AWS 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.

If you implement organization-level AWS CloudTrail trails outside of AWS Control Tower, you can use them with AWS Control Tower. However, if you have organization-level trails, you will incur duplicate charges, because AWS Control Tower requires account-level trails. We do not recommend setting up organization-level trails, unless you have a specific requirement, because AWS Control Tower performs their function by setting up CloudTrail trails for you in every account.
Setting up

Before you use AWS Control Tower for the first time, complete the following tasks:

1. Sign up for AWS (p. 16)
2. Create an IAM User (p. 16)

These tasks create an AWS account and an IAM user with administrator privileges for the account. For information on additional setup tasks specifically for AWS Control Tower, see Getting started with AWS Control Tower (p. 19).

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.

To create 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.

Note your AWS account number, because you need it for the next task.

Create an IAM User

Services in AWS, such as AWS Control Tower, require that your user account must provide credentials, so that the service can determine whether you have permission to utilize its resources. AWS recommends that you don't make requests to other services from the root user credentials of your AWS account. Instead, create an AWS Identity and Access Management (IAM) user and grant that user full access. We call these full-access users administrators.

You can use the administrator credentials, instead of AWS account root user credentials of your account, to interact with AWS and perform tasks, such as create users and grant them the appropriate permissions. For more information, see Root Account Credentials vs. IAM User Credentials in the AWS General Reference and IAM Best Practices in the IAM User Guide.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM Management Console.

To create an administrator user for yourself and add the user to an administrators group (console)

1. Sign in to the IAM console as the account owner by choosing Root user and entering your AWS account email address. On the next page, enter your password.
Note
We strongly recommend that you adhere to the best practice of using the Administrator IAM user that follows and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.

2. In the navigation pane, choose Users and then choose Add users.
3. For User name, enter Administrator.
4. Select the check box next to AWS Management Console access. Then select Custom password, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to User must create a new password at next sign-in to allow the new user to reset their password after they sign in.
6. Choose Next: Permissions.
7. Under Set permissions, choose Add user to group.
8. Choose Create group.
9. In the Create group dialog box, for Group name enter Administrators.
10. Choose Filter policies, and then select AWS managed - job function to filter the table contents.
11. In the policy list, select the check box for AdministratorAccess. Then choose Create group.

Note
You must activate IAM user and role access to Billing before you can use the AdministratorAccess permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.

12. Back in the list of groups, select the check box for your new group. Choose Refresh if necessary to see the group in the list.
13. Choose Next: Tags.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM entities in the IAM User Guide.
15. Choose Next: Review to see the list of group memberships to be added to the new user. When you are ready to proceed, choose Create user.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access management and Example policies.

To sign in as this new IAM user, first sign out of the AWS Management Console. Then use the following URL, where your_aws_account_id is your AWS account number without the hyphens (for example, if your AWS account number is 1234-5678-9012, your AWS account ID is 123456789012).

https://your_aws_account_id.signin.aws.amazon.com/console/

Enter the IAM user name and password that you just created. When you're signed in, the navigation bar displays your_user_name@your_aws_account_id.

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. To do so, from the IAM dashboard, choose Create Account Alias and enter an alias, such as your company name. To sign in after you create an account alias, use the following URL.

https://your_account_alias.signin.aws.amazon.com/console/

To verify the sign-in link for IAM users for your account, open the IAM console and check under AWS Account Alias on the dashboard.
Set up MFA

Because of the nature of AWS Control Tower, we strongly recommend that you enable multi-factor authentication (MFA) for your management account. For more information, see Enable MFA on the AWS Account Root User in the IAM User Guide.

Next Step

Getting started with AWS Control Tower (p. 19)
Getting started with AWS Control Tower

This getting started procedure is for AWS Control Tower central cloud administrators. Use 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 has a prerequisite and four steps.

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. 26).
- 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 Single Sign-On (AWS SSO) customers

- If AWS Single Sign-On (AWS SSO) is already set up, the AWS Control Tower home Region must be the same as the AWS SSO Region.
- AWS SSO can be installed only in the management account of an organization.
- Three options apply to your SSO directory, based on the identity source you choose in SSO:
  - **AWS SSO User Store**: If SSO for AWS Control Tower is set up with AWS SSO, AWS Control Tower creates groups in the SSO directory and provisions access to these groups, for the user you select, for member accounts.
  - **Active Directory**: If SSO for AWS Control Tower is set up with Active Directory, AWS Control Tower does not manage the SSO directory. It does not assign users or groups to new AWS accounts.
Considerations for AWS Config and AWS CloudTrail customers

- **External Identity Provider**: If AWS SSO for AWS Control Tower is set up with an external identity provider (IdP), AWS Control Tower creates groups in the SSO directory and provisions access to these groups for the user you select for member accounts. You can specify an existing user from your external IdP in Account Factory during account creation, and AWS Control Tower gives this user access to the newly vended account when it synchronizes users of the same name between SSO and the external IdP. You can also create groups in your external IdP to match the names of the default groups in AWS Control Tower. When you assign users to these groups, these users will have access to your enrolled accounts.

For more information about working with AWS SSO and AWS Control Tower see Things to Know About SSO Accounts and AWS Control Tower (p. 230)

Considerations for AWS Config and AWS CloudTrail customers

- The AWS account cannot have trusted access enabled in the organization management account for either AWS Config or AWS CloudTrail. For information about how to disable trusted access, see the AWS Organizations documentation on how to enable or disable trusted access.
- If you have an existing AWS Config recorder, delivery channel, or aggregation setup in any existing accounts that you plan to enroll in AWS Control Tower, you must modify or remove these configurations before you start enrolling the accounts, after your landing zone is set up. This pre-check doesn't apply to the AWS Control Tower management account during landing zone launch. For more information, see Enroll accounts that have existing AWS Config resources (p. 112).
- If you are running ephemeral workloads from accounts in AWS Control Tower, you may see an increase in costs associated with AWS Config. Contact your AWS account representative for more specific information about managing these costs.
- 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 the account, you may see duplicate charges unless you delete the existing trail for the account before you enroll it in AWS Control Tower. For information about organization-level trails and AWS Control Tower, see Pricing (p. 15).

**Note**
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.

Requirements for your shared account email addresses

If you're setting up your landing zone in a new AWS account, for information on creating your account and your IAM administrator, see Setting up (p. 16).

To set up your landing zone, 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. 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 created 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 AWS SSO user with the appropriately scoped permissions.

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 are "one-way doors" that 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. 38).

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

### Configuration choices that cannot be undone

- You cannot change your home Region after you've set up your landing zone.

- If you’re provisioning Account Factory accounts with VPCs, VPC CIDRs can’t be changed after they are created.

## 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. 28).

**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. To configure and launch your landing zone, 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 1. Review pricing 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.

Step 2. 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. 38). 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 3. Configure your shared accounts 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, although you can choose customized names for them during setup.

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
Step 3. Configure your shared accounts and encryption

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, and you must supply a unique email address for each account. Remember that the email addresses must not already have associated AWS accounts.

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

1. At the console, select 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. Select 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 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.

**To make the key's policy update**

To use a KMS key with AWS Control Tower, you must make a specific policy update to the key. At minimum, the KMS key must have permissions that allow AWS CloudTrail and AWS Config to use the chosen KMS key.

**Make the required policy update**

1. Navigate to the AWS KMS console at [https://console.aws.amazon.com/kms](https://console.aws.amazon.com/kms)
2. Select **Customer managed keys** on the left
3. In the table, select the key you wish to edit, or select **Create a key** from the upper right
4. Under the section called **Key policy**, make sure you can see the policy and edit it. You may need to select **Switch to policy view** on the right.
You can copy and paste the following example policy statement. Alternatively, for an existing key, you can ensure that your KMS key has these minimum permissions by adding them to your own existing policy. You can add these lines as a group in a single JSON statement, or if you prefer, you can incorporate them line by line into your policy's other statements.

```
{
  "Sid": "Allow CloudTrail and AWS Config to encrypt/decrypt logs",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "cloudtrail.amazonaws.com",
      "config.amazonaws.com"
    ]
  },
  "Action": [
    "kms:GenerateDataKey",
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

The AWS Key Management Service (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.

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

**Step 4. 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 AWS SSO users and groups with specific roles and permissions. For recommendations, see Recommendations for setting up groups, roles, and policies (p. 29).
- 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. 117).
- To assure Compliance Validation for AWS Control Tower (p. 245), 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 guardrails, and to keep your landing zone up-to-date. For more information, see Configuration update management in AWS Control Tower (p. 35).
- If you encounter issues while using AWS Control Tower, see Troubleshooting (p. 278).
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.

Limitations in AWS Control Tower

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

- 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 5 SCPs per OU applies to OUs in your AWS Control Tower landing zone.
- Existing OUs with over 300 accounts cannot be registered or re-registered in AWS Control Tower.

For information about how to increase certain AWS Control Tower service quotas with an automated request method, view this video: Automate Service Limit Increases. When provisioning new accounts in this environment, you can use lifecycle events to trigger automated requests for service limit increases in specified AWS Regions. The video also shows how to automate enrollment of new accounts into Enterprise support for your organization.

Quotas for Integrated Services

Each AWS service has its own quotas and limits. You can find the quotas for each service in its documentation. For more information, see the related links:

- **AWS CloudFormation** – AWS CloudFormation Quotas
- **AWS CloudTrail** – Quotas in AWS CloudTrail
- **Amazon CloudWatch** – CloudWatch Quotas
- **AWS Config** – AWS Config Quotas
- **AWS Identity and Access Management** – Quotas for IAM Entities and Objects
- **AWS Lambda** – AWS Lambda Quotas
- **AWS Organizations** – Quotas for AWS Organizations
- **Amazon Simple Storage Service** – Bucket Restrictions and Quotas
- **AWS Service Catalog** – AWS Service Catalog Default Service Quotas
- **AWS Single Sign-On** – Quotas in AWS SSO
- **Amazon Simple Notification Service** – Amazon Simple Notification Service (Amazon SNS) Quotas
- **AWS Step Functions** – Quotas
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 guardrails 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 guardrails 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.
- **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.
- **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.

Actions taken by CloudTrail landing zone administrators are logged in the landing zone management account. Actions taken by member account administrators and users are logged in the shared log archive account.
You can view a summary table of AWS Control Tower events in the Activities page.

Explaining Preventive Guardrails

A preventive guardrail ensures that your organization's accounts maintain compliance with your corporate policies. The status of a preventive guardrail is either enforced or not-enabled. A preventive guardrail prevents policy violations by using service control policies (SCPs). In comparison, a detective guardrail 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 guardrails 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 Single Sign-On user.

To show preventive guardrails for users with a need to know

2. From the left navigation, choose Organizational units.
3. From the table, choose the name of one of the OUs for which your user needs information about the applicable guardrails.
4. Note the name of the OU and the guardrails 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 guardrails and their functions, see Guardrails in AWS Control Tower (p. 173).

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 AWS Service Catalog portfolio, or to create a custom VPC that runs in only 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 AWS 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. 235).

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 guardrails, and SCPs, and access to AWS Config, for managing detective guardrails, 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. 74).

Guidance for creating and modifying AWS Control Tower resources

We recommend the following 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.
AWS Control Tower User Guide
Sign in as a Root User

- 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 (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 those Regions. Instead, rely on the AWS Control Tower Region deny capability, as shown in the guardrail, Deny access to AWS based on the requested AWS Region (p. 210).
- 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.
- In general, AWS Control Tower performs a single action at a time, which must be completed before another action can begin. For example, if you attempt to provision an account while the process of enabling a guardrail is already in operation, account provisioning will fail. The only concurrent action allowed in AWS Control Tower is deployment of detective guardrails. See Concurrent detective guardrail deployment (p. 176).
- Keep an active AWS Config recorder. If you delete your Config recorder, detective guardrails cannot detect and report drift. Non-compliant resources may be reported as Compliant due to insufficient information.
- Do not delete the AWSControlTowerExecution role from your member accounts, even in unenrolled accounts. If you do, you will not be able to enroll these accounts with AWS Control Tower, or register their immediate parent OUs.

Tips about accounts and OUs

- We recommend that you keep each registered OU to a maximum of 300 accounts, so that you can update those accounts with the Re-register OU capability whenever account updates are required, such as when you configure new Regions for governance.
- To reduce the time required when registering an OU, we recommend that you keep the number of accounts per OU to around 150, even though the limit is 300 accounts per OU. As a general rule, the time required to register an OU increases according to the number of Regions in which your OU is operating, multiplied by the number of accounts in the OU.
- As an estimate, an OU with 150 accounts requires approximately 2 hours to register and enable guardrails, and about 1 hour to re-register. Also, an OU that has many guardrails takes longer to register than an OU with few guardrails.
- One concern about allowing a longer timeframe for registering an OU is that this process blocks other actions. Some customers are comfortable allowing longer times to register or re-register an OU, because they prefer to allow more accounts in each OU.

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 Service Catalog (p. 119).
To change or enable your AWS Support plan.
To close an AWS Account.
For more information about actions that require root login credentials, please see AWS Tasks that Require AWS Root Login Credentials.

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.

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.

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
- Do not 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 guardrails 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. 151).
- 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.
AWS Single Sign-On Guidance

- For specific information about how AWS Control Tower works with SSO based on your identity source, see Considerations for AWS Single Sign-On (SSO) customers in the Prerequisites section of the Getting Started page of this User Guide.
- For additional information about how the behavior of AWS Control Tower interacts with AWS SSO and different identity sources, refer to Considerations for Changing Your Identity Source in the AWS SSO documentation.
- See Managing Users and Access Through AWS Single Sign-On (p. 229) for more information about working with AWS Control Tower and AWS SSO.

Account Factory Guidance

- When you use Account Factory to provision new accounts in AWS Control Tower, do not define TagOptions, enable notifications, or create a provisioned product plan. Doing so can result in a failure to provision a new account.
- If you are authenticated as an IAM user when you provision accounts in Account Factory or when you use the Enroll account feature, be sure the IAM user is added to the AWS Service Catalog portfolio so that it has the correct permissions. Otherwise, you may receive an error message from AWS Service Catalog that is difficult to understand. Common causes for this type of error are given in the Troubleshooting guide. In particular, refer to the section entitled No Launch Paths Found Error (p. 283).
- Remember that only one account 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 AWS CloudWatch event notifications. For example, this topic informs you if a guardrail 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.

For more information about SNS topics and compliance, see Drift prevention and notification (p. 179).

Guidance for KMS keys

AWS Control Tower works with AWS Key Management Service (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.

The AWS Key Management Service (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 To make the key’s policy update (p. 23).

AWS Control Tower and VPCs

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.

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. By default, each Availability Zone is assigned one public subnet and two private subnets. Therefore, each AWS Control Tower VPC contains nine subnets by default, divided into three Availability Zones.
- Each of the nine 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 nine subnets within your AWS Control Tower VPC can communicate with each other in an unrestricted manner.

Important

If you provision Account Factory accounts with VPC internet access settings enabled, that Account Factory setting overrides the guardrail Disallow internet access for an Amazon VPC instance managed by a customer (p. 211). 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. 272).

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.

Note

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.
CIDR and Peering for VPC and AWS Control Tower

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. 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. 277).

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.

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.
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>2.8</td>
<td>2-10-2022</td>
<td>Landing zone version 2.8 (p. 290)</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>

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. Accounts 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 2.6, 2.7, and so forth.
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, start at Step 3.

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 (must be used for an OU that contains over 300 accounts). From the navigation pane, choose Accounts.
4. Choose Enroll account to open the Account Factory product.
5. From the navigation pane, choose Provisioned products list.
6. For each account listed, perform the following steps to update all your member accounts:
   a. From the menu for the account, choose Provisioned product details.
   b. Make a note of the following parameters:
      • SSOUserEmail (Available in provisioned product details)
      • AccountEmail (Available in provisioned product details)
      • SSOUserName (Available in SSO)
      • SSOUserLastName (Available in SSO)
      • AccountName (Available in SSO)
   c. From Actions, choose Update.
   d. Choose the radio button next to the Version of the product you want to update, and choose Next.
   e. Provide the parameter values that were mentioned previously. For ManagedOrganizationalUnit choose the OU that the account is in. You can find this information in the AWS Control Tower console, under Accounts.
   f. Choose Next.
   g. Review your changes, and then choose Update. This process can take a few minutes per account.

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

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. 151) and Detect and resolve drift in AWS Control Tower (p. 148).

Provisioning and updating accounts using script automation

You can provision or update individual accounts by using the API framework of AWS Service Catalog and the AWS CLI to update the accounts in a batch process. You'd call the `UpdateProvisionedProduct` API of AWS 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.

You must wait for each 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 AWS Service Catalog APIs (p. 274).

**Note**

The Video Walkthrough (p. 277) is designed for automated account provisioning, but the steps also apply to account updating. Use the `UpdateProvisionedProduct` API instead of the `ProvisionProduct` API.

A further step of automation 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.

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 guardrails 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 guardrails 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.

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. 164).
- 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 guardrails 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.

**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 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 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 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.

2. 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:w:wad4a567d8bfd9a1234eb4e56\"
   }
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.
Automating tasks in AWS Control Tower

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

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

The Related information (p. 286) 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.

Automated account provisioning and updating

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 (p. 124).

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 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 SSO 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.
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 guardrails. 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. 148).
- 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 guardrail to Enabled or Not enabled, and control user access to most AWS services in ungoverned AWS Regions.

Customize by adding optional guardrails

- Strongly recommended and elective guardrails are optional, which means that you can customize the level of enforcement for your landing zone by choosing which ones to enable. Optional guardrails (p. 176) are not enabled by default.
- The optional Guardrails that enhance data residency protection (p. 209) allow you to customize the Regions in which you store and allow access to your data.
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 (p. 124).
- You can customize your AWS Control Tower landing zone with Customizations for AWS Control Tower (p. 49) (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.

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. 252), 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.

Source code is available in a GitHub repository.

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 AWS Service Catalog and AWS Control Tower customizations.
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 section provides architectural considerations and configuration steps for deploying Customizations for AWS Control Tower (CfCT). It includes a link to the AWS 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.
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. 19).

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. 57) (p. 57).

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. 61).

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. 59).

**The AWS Control Tower lifecycle event workflow**

When a new account is created in AWS Control Tower, a *lifecycle event* (p. 252) 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).

Amazon Simple Storage Service

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

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. 59).

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.

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 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. 57). 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. 59)

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.

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.

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.

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. 57).

**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. 59).

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. 59).

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. 54).

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. 57).
Template and source code

Customizations for AWS Control Tower (CfCT) is deployed automatically in your management account, by a AWS CloudFormation template. You can review the AWS CloudFormation template and download it before deployment.

The custom-control-tower-initiation.template: This template deploys an AWS CodePipeline, AWS CodeBuild projects, AWS Step Functions, AWS Lambda functions, an Amazon EventBridge event rule, an Amazon Simple Queue Service queue, and an Amazon Simple Storage Service bucket with a sample configuration package. You can customize the template based on your specific requirements.

Source code repository

You can visit our GitHub repository 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. 52). 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. 19).

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. 54)

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

Step 2. Create a custom package (p. 57)

- Create a custom configuration package.

Step 1. Launch the stack

The automated 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. 51).
1. Sign in to the AWS Management Console, then choose the link that follows, to launch the custom-control-tower-initiation AWS CloudFormation template.

**Launch Customizations for AWS Control Tower (CfCT)**

You can also download the template as a starting point for your own implementation.

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 Amazon S3 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.

<table>
<thead>
<tr>
<th>Pipeline Configuration</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pipeline Approval Stage</strong></td>
<td>No</td>
<td>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. 59).</td>
</tr>
<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**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Existing CodeCommit Repository?</strong></td>
<td>No</td>
<td>Choose whether to use an existing CodeCommit Git repository. If you choose Yes, you must set the CodePipeline Source parameter to AWS CodeCommit.</td>
</tr>
<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 enter the repository’s URL in the CodeCommit Repository URL field.</td>
</tr>
</tbody>
</table>

55
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 will create AWS Identity and Access Management (IAM) resources.

9. Choose Create 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 CREATE_COMPLETE in approximately 15 minutes.
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. 59).

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

Update the stack

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

1. Sign in to the AWS CloudFormation console, select your existing Customizations for AWS Control Tower (CfCT) CloudFormation stack, and select Update.
2. Select Replace current template.
3. Under Specify template:
   a. Select Amazon S3 URL.
   b. Copy the link of the latest template.
   c. Paste the link in the Amazon S3 URL box.
   d. Verify that the correct template URL shows in the Amazon S3 URL text box and choose Next. Choose Next again.
4. Under Parameters, review the parameters for the template and modify them as necessary. Refer to Step 1. Launch the stack (p. 54) for details about the parameters.
5. Choose Next.
6. On the Configure stack options page, choose Next.
7. 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.
8. Choose View change set and verify the changes.
9. 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.

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. 59).
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.

Collection of operational metrics

Customizations for AWS Control Tower (CfCT) includes an option to send anonymous operational metrics to AWS. We use this data to understand how customers use this capability, along with 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 that AWS owns the data gathered through this survey. Data collection is subject to the AWS Privacy Policy. To opt out of this feature, complete one of the following tasks:

- Modify the AWS CloudFormation template mapping section as follows:

<table>
<thead>
<tr>
<th>AnonymousData:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SendAnonymousData:</td>
</tr>
<tr>
<td>Data: Yes</td>
</tr>
</tbody>
</table>

  to

<table>
<thead>
<tr>
<th>AnonymousData:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SendAnonymousData:</td>
</tr>
<tr>
<td>Data: No</td>
</tr>
</tbody>
</table>
After CfCT is deployed, find the /org/primary/metrics_flag SSM parameter key in the Parameter Store console and set the value to No.

CfCT customization guide

This topic is intended for IT infrastructure architects, administrators, DevOps professionals, systems integrators, or independent software vendors who want to customize and extend the AWS Control Tower environment for their company or customers.

This guide provides detailed, step-by-step information to help you customize and extend the AWS Control Tower environment with the CfCT customization package. You'll find information about the custom configuration package structure, manifest schema, configuration templates, and pipeline deployment stages.

To deploy and configure Customizations for AWS Control Tower (CfCT), you will deploy and process a configuration package through AWS CodePipeline. The following sections describe this 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:

custom-control-tower-configuration-accountID-region.

**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 Edit a pipeline in CodePipeline in the AWS CodePipeline User Guide.

The 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 pipeline stages (p. 60).

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.

**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 custom-control-tower-configuration.zip without the underscore (_) in order to trigger the deployment in AWS CodePipeline.
AWS CodePipeline stages

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

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.

2. **Build stage**
   
   The build stage requires AWS CodeBuild to validate the contents of the configuration package. These checks include testing the `manifest.yaml` file syntax and schema, along with all AWS CloudFormation templates included in the package or remotely hosted, using AWS CloudFormation `validate-template` and `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.

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 **Pipeline Approval Stage** parameter to **Yes** when you launch the stack.

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 CloudFoundation 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. 71).

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 entries for the `resources` section of the manifest file in detail, where you'll define the resources required for your customizations.

This section of the manifest file begins at the keyword `resources`, and it continues to the end of the file.

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

The stack instances are deployed at the account level rather than the OU level. The SCPs are deployed at the OU level. Refer to [Build Your Own Customizations](p. 66) for details.

The code example that follows is a pseudo template, which summarizes the possible entries available for the `resources` section in the manifest file.

```yaml
resources: # List of resources
  - name: [String]
    resource_file: [String] [Local File Path, S3 URL]```
Define a custom configuration

```yaml
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: (1) the relative location to the manifest file, (2) an Amazon S3 URL that points to an AWS CloudFormation template or AWS Organizations service control policy in JSON for creating AWS CloudFormation resources or SCPs, respectively.

- **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 URL

   ```yaml
   resources:
   - name: SecurityRoles
     resource_file: s3://my-bucket/[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.

```json
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**.

• **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.

```yaml
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`.

```yaml
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 an OU value 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 **AWSControlTowerExecutionRole** 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.
### 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: 
  ...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).
- name: test-preventive-guardrails
description: To prevent from deleting or disabling resources in member accounts
resource_file: policies/preventive-guardrails.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 block 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.

```json
{
  "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.

```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
---
```

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
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 parameter 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. 68).
The ‘alfred’ helper and the AWS CloudFormation parameter files

CfCT provides a mechanism to get the value for an SSM Parameter Store key defined in the AWS CloudFormation parameter JSON file (.json). This mechanism allows you to use values stored in the parameter store without updating the AWS CloudFormation template.

For example, suppose that AWS CloudFormation stack 1 creates an Amazon VPC and subnets in an availability zone. The VPC ID and subnet ID must be passed into AWS CloudFormation stack 2 as parameter values.

You can store the VPC ID and the subnet ID into the SSM parameter store, using the AWS::SSM::Parameter resource in stack 1. As shown in the example that follows, the alfred helper can get the values from the parameter store and pass them into the StackSet state machine, as input.

**AWS CloudFormation stack 1 snippet:**

```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
```

The following example shows the parameters for AWS CloudFormation stack 2 specified in the 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']
```

Starting with version 2.1, you can use a list of alfred_ssm properties under parameter_value to support the CommaDelimitedList type of AWS CloudFormation parameter. For example:

```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

The 2.2.0 release of Customizations for AWS Control Tower (CfCT) introduces a new manifest schema version, 2021-03-15. This version simplifies and aligns with related AWS service APIs. It allows you to maintain a single file (manifest.yaml) that manages supported resources (AWS CloudFormation templates and SCPs) through decoupled DevOps workflows.

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

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

Deprecated properties in manifest version 2021-03-15:

<table>
<thead>
<tr>
<th>property</th>
</tr>
</thead>
<tbody>
<tr>
<td>organization_policies</td>
</tr>
<tr>
<td>policy_file</td>
</tr>
<tr>
<td>apply_to_accounts_in_ou</td>
</tr>
<tr>
<td>cloudformation_resources</td>
</tr>
<tr>
<td>template_file</td>
</tr>
<tr>
<td>deploy_to_account</td>
</tr>
<tr>
<td>deploy_to_ou</td>
</tr>
<tr>
<td>ssm_parameters</td>
</tr>
</tbody>
</table>

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 policies

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.
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. Refer to Permissions Required to Use the AWS Control Tower Console (p. 241) for information about the roles required by AWS Control Tower. For general information about IAM and roles in AWS, see the IAM roles topic in the AWS IAM User Guide.

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) guardrails, 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 guardrails. 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. 103).

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. 110).

How the `AWSControlTowerExecution` role works with OUs:

The `AWSControlTowerExecution` role ensures that your selected AWS Control Tower guardrails apply automatically to every individual account, in each OU, in your organization, as well as to every new account you create in AWS Control Tower. As a result:

• You can provide compliance and security reports more easily, based on the auditing and logging features embodied by AWS Control Tower guardrails.

• 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.

To summarize, the `AWSControlTowerExecution` role and its associated policy gives you flexible control of security and compliance across your entire organization. Therefore, breaches of security or protocol are less likely to occur.

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.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    
```
Optional conditions for your role trust relationships

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": ["controltower.amazonaws.com"
    ],
    "Action": "sts:AssumeRole",
    "Condition": {
        "ArnEquals": {
            "aws:SourceArn": "arn:aws:controltower:us-west-2:012345678901:*
        }
    }
}
```

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.

```
{
    "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. 240).

For general information about permission policies in AWS Control Tower see [Managing Access to Resources](p. 238).

**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. 94), we recommend that you add the aws:SourceArn condition to these roles with aws:PrincipalOrgID, which validates that the principal accessing the resource belongs to an account in the correct AWS organization.

**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. 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.

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

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

**Assumed by:** AWS Config

**AWS managed policies:**
- ReadOnlyAccess
- AWS_ConfigRole

The following JSON artifact shows the role.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "a4b:Get*",
                "a4b:List*",
                "a4b:Search*",
                "access-analyzer:GetAccessPreview",
                "access-analyzer:GetAnalyzedResource",
                "access-analyzer:GetAnalyzer",
                "access-analyzer:GetArchiveRule",
                "access-analyzer:GetFinding",
```
"access-analyzer:GetGeneratedPolicy",
"access-analyzer:ListAccessPreviewFindings",
"access-analyzer:ListAccessPreviews",
"access-analyzer:ListAnalyzedResources",
"access-analyzer:ListAnalyzers",
"access-analyzer:ListArchiveRules",
"access-analyzer:ListFindings",
"access-analyzer:ListPolicyGenerations",
"access-analyzer:ListTagsForResource",
"access-analyzer:ValidatePolicy",
"acm-pca:Describe*",
"acm-pca:Get*",
"acm-pca:List*",
"acm:Get*",
"acm:List*",
"airflow:ListEnvironments",
"airflow:ListTagsForResource",
"amplify:GetApp",
"amplify:GetBranch",
"amplify:GetDomainAssociation",
"amplify:GetJob",
"amplify:ListApps",
"amplify:ListBranches",
"amplify:ListDomainAssociations",
"amplify:ListJobs",
"apigateway:GET",
"appconfig:GetApplication",
"appconfig:GetConfiguration",
"appconfig:GetConfigurationProfile",
"appconfig:GetDeployment",
"appconfig:GetDeploymentStrategy",
"appconfig:GetEnvironment",
"appconfig:GetHostedConfigurationVersion",
"appconfig:ListApplications",
"appconfig:ListConfigurationProfiles",
"appconfig:ListDeployments",
"appconfig:ListDeploymentStrategies",
"appconfig:ListEnvironments",
"appconfig:ListHostedConfigurationVersions",
"appconfig:ListTagsForResource",
"application-autoscaling:Describe*",
"applicationinsights:Describe*",
"applicationinsights:List*",
"appmesh:Describe*",
"appmesh:List*",
"appstream:Describe*",
"appstream:List*",
"appsync:Get*",
"appsync:List*",
"aps:DescribeAlertManagerDefinition",
"aps:DescribeRuleGroupsNamespace",
"aps:DescribeWorkspace",
"aps:GetAlertManagerSilence",
"aps:GetAlertManagerStatus",
"aps:GetLabels",
"aps:GetMetricMetadata",
"aps:GetSeries",
"aps:ListAlerts",
"aps:ListAlertManagerAlerts",
"aps:ListAlertManagerAlertGroups",
"aps:ListAlertManagerReceivers",
"aps:ListAlertManagerSilences",
"aps:ListRules",
"aps:ListRuleGroupsNamespaces",
"aps:ListTagsForResource"
"aps:ListWorkspaces",
"aps:QueryMetrics",
"athena:Batch**",
"athena:Get*",
"athena:List*",
"auditmanager:GetAccountStatus",
"auditmanager:GetAssessment",
"auditmanager:GetAssessmentFramework",
"auditmanager:GetAssessmentReportUrl",
"auditmanager:GetChangeLogs",
"auditmanager:GetControl",
"auditmanager:GetDelegations",
"auditmanager:GetEvidence",
"auditmanager:GetEvidenceByEvidenceFolder",
"auditmanager:GetEvidenceFolder",
"auditmanager:GetEvidenceFoldersByAssessment",
"auditmanager:GetEvidenceFoldersByAssessmentControl",
"auditmanager:GetOrganizationAdminAccount",
"auditmanager:GetServicesInScope",
"auditmanager:GetSettings",
"auditmanager:ListAssessmentFrameworks",
"auditmanager:ListAssessmentReports",
"auditmanager:ListAssessments",
"auditmanager:ListControls",
"auditmanager:ListKeywordsForDataSource",
"auditmanager:ListNotifications",
"auditmanager:ListTagsForResource",
"auditmanager:ValidateAssessmentReportIntegrity",
"autoscaling-plans:Describe*",
"autoscaling-plans:GetScalingPlanResourceForecastData",
"aws-portal:View**",
"backup:Describe*",
"backup:Get*",
"backup:List*",
"batch:Describe**",
"batch:List*",
"braket:GetDevice",
"braket:GetQuantumTask",
"braket:SearchDevices",
"braket:SearchQuantumTasks",
"budgets:Describe**",
"budgets:View**",
"cassandra:Select",
"ce:DescribeCostCategoryDefinition",
"ce:DescribeNotificationSubscription",
"ce:DescribeReport",
"ce:GetAnomalies",
"ce:GetAnomalyMonitors",
"ce:GetAnomalySubscriptions",
"ce:GetCostAndUsage",
"ce:GetCostAndUsageWithResources",
"ce:GetCostCategories",
"ce:GetCostForecast",
"ce:GetDimensionValues",
"ce:GetPreferences",
"ce:GetReservationCoverage",
"ce:GetReservationPurchaseRecommendation",
"ce:GetReservationUtilization",
"ce:GetRightsizingRecommendation",
"ce:GetSavingsPlansCoverage",
"ce:GetSavingsPlansPurchaseRecommendation",
"ce:GetSavingsPlansUtilization",
"ce:GetSavingsPlansUtilizationDetails",
"ce:GetTags",
"ce:GetUsageForecast",
"ce:ListCostCategoryDefinitions",
"chatbot:Describe**",
"chatbot:Get**",
"chime:Get**",
"chime:List**",
"chime:Retrieve**",
"chime:Search**",
"chime:Validate**",
"cloud9:Describe**",
"cloud9:List**",
"clouddirectory:BatchRead",
"clouddirectory:Get**",
"clouddirectory:List**",
"clouddirectory:LookupPolicy",
"cloudformation:Describe**",
"cloudformation:Detect**",
"cloudformation:Estimate**",
"cloudformation:Get**",
"cloudformation:List**",
"cloudfront:DescribeFunction",
"cloudfront:Get**",
"cloudfront:List**",
"cloudhsm:Describe**",
"cloudhsm:Get**",
"cloudhsm:List**",
"cloudsearch:Describe**",
"cloudsearch:List**",
"cloudtrail:Describe**",
"cloudtrail:Get**",
"cloudtrail:List**",
"cloudtrail:LookupEvents",
"cloudwatch:Describe**",
"cloudwatch:Get**",
"cloudwatch:List**",
"codeartifact:DescribeDomain",
"codeartifact:DescribePackageVersion",
"codeartifact:DescribeRepository",
"codeartifact:GetAuthorizationToken",
"codeartifact:GetDomainPermissionsPolicy",
"codeartifact:GetPackageVersionAsset",
"codeartifact:GetPackageVersionReadme",
"codeartifact:GetRepositoryEndpoint",
"codeartifact:GetRepositoryPermissionsPolicy",
"codeartifact:GetRepositoriesInDomain",
"codeartifact:ListDomains",
"codeartifact:ListPackages",
"codeartifact:ListPackageVersionAssets",
"codeartifact:ListPackageVersionDependencies",
"codeartifact:ListPackageVersions",
"codeartifact:ListRepositories",
"codeartifact:ListTagsForResource",
"codeartifact:ReadFromRepository",
"codebuild:BatchGet**",
"codebuild:DescribeCodeCoverages",
"codebuild:DescribeTestCases",
"codebuild:List**",
"codecommit:BatchGet**",
"codecommit:Describe**",
"codecommit:Get**",
"codecommit:GitPull",
"codecommit:List**",
"codedeploy:BatchGet**",
"codedeploy:Get**",
"codedeploy:List**",
"codeguru-profiler:Describe**",
"codeguru-profiler:Get**", 79
"codeguru-profiler:List*",
"codeguru-reviewer:Describe*",
"codeguru-reviewer:Get*",
"codeguru-reviewer:List*",
"codepipeline:Get*",
"codepipeline:List*",
"codestar-connections:GetConnection",
"codestar-connections:GetHost",
"codestar-connections:ListConnections",
"codestar-connections:ListHosts",
"codestar-connections:ListTagsForResource",
"codestar-notifications:describeNotificationRule",
"codestar-notifications:listEventTypes",
"codestar-notifications:listNotificationRules",
"codestar-notifications:listTagsForResource",
"codestar-notifications:ListTargets",
"codestar:Describe*",
"codestar:Get*",
"codestar:List*",
"codestar:Verify*",
"cognito-identity:Describe*",
"cognito-identity:GetCredentialsForIdentity",
"cognito-identity:GetIdentityPoolRoles",
"cognito-identity:GetOpenIdToken",
"cognito-identity:GetOpenIdTokenForDeveloperIdentity",
"cognito-identity:List*",
"cognito-identity:Lookup*",
"cognito-idp:AdminGet*",
"cognito-idp:AdminList*",
"cognito-idp:Describe*",
"cognito-idp:Get*",
"cognito-idp:List*",
"cognito-sync:Describe*",
"cognito-sync:Get*",
"cognito-sync:List*",
"cognito-sync:QueryRecords",
"compute-optimizer:DescribeRecommendationExportJobs",
"compute-optimizer:GetAutoScalingGroupRecommendations",
"compute-optimizer:GetEBSVolumeRecommendations",
"compute-optimizer:GetEC2InstanceRecommendations",
"compute-optimizer:GetEC2RecommendationProjectedMetrics",
"compute-optimizer:GetEnrollmentStatus",
"compute-optimizer:GetEnrollmentStatusesForOrganization",
"compute-optimizer:GetLambdaFunctionRecommendations",
"compute-optimizer:GetRecommendationSummaries",
"config:BatchGetAggregateResourceConfig",
"config:BatchGetResourceConfig",
"config:Deliver*",
"config:Describe*",
"config:Get*",
"config:List*",
"config:SelectAggregateResourceConfig",
"config:SelectResourceConfig",
"connect:Describe*",
"connect:GetFederationToken",
"connect:List*",
"dataexchange:Get*",
"dataexchange:List*",
"datapipeline:Describe*",
"datapipeline:EvaluateExpression",
"datapipeline:Get*",
"datapipeline:List*",
"datapipeline:QueryObjects",
"datapipeline:Validate*",
"datasync:Describe*",
"datasync:List*"
"dax:BatchGetItem",
"dax:Describe*",
"dax:GetItem",
"dax:ListTags",
"dax:Query",
"dax:Scan",
"deepcomposer:GetComposition",
"deepcomposer:GetModel",
"deepcomposer:GetSampleModel",
"deepcomposer:ListCompositions",
"deepcomposer:ListModels",
"deepcomposer:ListSampleModels",
"deepcomposer:ListTrainingTopics",
"detective:Get*",
"detective:List*",
"detective:SearchGraph",
"devicefarm:Get**",
"devicefarm:List*",
"devops-guru:DescribeAccountHealth",
"devops-guru:DescribeAccountOverview",
"devops-guru:DescribeAnomaly",
"devops-guru:DescribeFeedback",
"devops-guru:DescribeInsight",
"devops-guru:DescribeResourceCollectionHealth",
"devops-guru:DescribeServiceIntegration",
"devops-guru:GetCostEstimation",
"devops-guru:GetResourceCollection",
"devops-guru:ListAnomaliesForInsight",
"devops-guru:ListEvents",
"devops-guru:ListInsights",
"devops-guru:ListNotificationChannels",
"devops-guru:ListRecommendations",
"devops-guru:SearchInsights",
"devops-guru:StartCostEstimation",
"directconnect:Describe*",
"discovery:Describe**",
"discovery:Get**",
"discovery:List*",
"dlm:Get*",
"dms:Describe*",
"dms:List*",
"dms:Test*",
"ds:Check*",
"ds:Describe*",
"ds:Get*",
"ds:List*",
"ds:Verify*",
"dynamodb:BatchGet*",
"dynamodb:Describe*",
"dynamodb:Get*",
"dynamodb:List*",
"dynamodb:Query",
"dynamodb:Scan",
"ec2:Describe*",
"ec2:Get*",
"ec2:SearchTransitGatewayRoutes",
"ec2messages:Get*",
"ecr-public:BatchCheckLayerAvailability",
"ecr-public:DescribeImages",
"ecr-public:DescribeImageTags",
"ecr-public:DescribeRepositories",
"ecr-public:GetAuthorizationToken",
"ecr-public:GetRegistryCatalogData",
"ecr-public:GetRepositoryCatalogData",
"ecr-public:GetRepositoryPolicy"
"ecr-public:ListTagsForResource",
"ecr:BatchCheck*",
"ecr:BatchGet*",
"ecr:Describe*",
"ecr:Get*",
"ecr:List*",
"ecs:Describe*",
"ecs:List*",
"eks:Describe*",
"eks:List*",
"elasticache:Describe*",
"elasticache:List*",
"elasticbeanstalk:Check*",
"elasticbeanstalk:Describe*",
"elasticbeanstalk:List*",
"elasticbeanstalk:Request*",
"elasticbeanstalk:Retrieve*",
"elasticbeanstalk:Validate*",
"elasticfilesystem:Describe*",
"elasticfilesystem:List*",
"elasticloadbalancing:Describe*",
"elasticmapreduce:Describe*",
"elasticmapreduce:GetBlockPublicAccessConfiguration",
"elasticmapreduce:List*",
"elasticmapreduce:View*",
"elastictranscoder:List*",
"elastictranscoder:Read*",
"elemental-appliances-software:Get*",
"elemental-appliances-software:List*",
"es:Describe*",
"es:EHttpGet",
"es:EHttpHead",
"es:Get*",
"es:List*",
"events:Describe*",
"events:List*",
"events:Test*",
"firehose:Describe*",
"firehose:List*",
"fis:GetAction",
"fis:GetExperiment",
"fis:GetExperimentTemplate",
"fis:GetListActions",
"fis:GetListExperiments",
"fis:GetListExperimentTemplates",
"fis:GetListTagsForResource",
"fms:GetAdminAccount",
"fms:GetAppsList",
"fms:GetComplianceDetail",
"fms:GetNotificationChannel",
"fms:GetPolicy",
"fms:GetProtectionStatus",
"fms:GetProtocolsList",
"fms:GetViolationDetails",
"fms:GetAppsLists",
"fms:GetListComplianceStatus",
"fms:GetListMemberAccounts",
"fms:GetListPolicies",
"fms:GetListProtocolsLists",
"fms:GetListTagsForResource",
"forecast:DescribeDataset",
"forecast:DescribeDatasetGroup",
"forecast:DescribeDatasetImportJob",
"forecast:DescribeForecast",
"forecast:DescribeForecastExportJob",
"forecast:DescribePredictor",
"forecast:DescribePredictorBacktestExportJob"
"forecast:GetAccuracyMetrics",
"forecast:ListDatasetGroups",
"forecast:ListDatasetImportJobs",
"forecast:ListDatasets",
"forecast:ListForecastExportJobs",
"forecast:ListForecasts",
"forecast:ListPredictorBacktestExportJobs",
"forecast:ListPredictors",
"forecast:QueryForecast",
"freertos:Describe*",
"freertos:List*",
"fsx:Describe*",
"fsx:List*",
"gamelift:Describe*",
"gamelift:Get*",
"gamelift:List*",
"gamelift:ResolveAlias",
"gamelift:Search*",
"glacier:Describe*",
"glacier:Get*",
"glacier:List*",
"globalaccelerator:Describe*",
"globalaccelerator:List*",
"glue:BatchGetDevEndpoints",
"glue:BatchGetJobs",
"glue:BatchGetPartition",
"glue:BatchGetTriggers",
"glue:BatchGetWorkflows",
"glue:CheckSchemaVersionValidity",
"glue:GetCatalogImportStatus",
"glue:GetClassifier",
"glue:GetClassifiers",
"glue:GetCrawler",
"glue:GetCrawlerMetrics",
"glue:GetCrawlers",
"glue:GetDatabase",
"glue:GetDatabases",
"glue:GetDataCatalogEncryptionSettings",
"glue:GetDataflowGraph",
"glue:GetDevEndpoint",
"glue:GetDevEndpoints",
"glue:GetJob",
"glue:GetJobBookmark",
"glue:GetJobRun",
"glue:GetJobRuns",
"glue:GetJobs",
"glue:GetMapping",
"glue:GetMLTaskRun",
"glue:GetMLTaskRuns",
"glue:GetMLTransform",
"glue:GetMLTransforms",
"glue:GetPartition",
"glue:GetPartitions",
"glue:GetPlan",
"glue:GetRegistry",
"glue:GetResourcePolicy",
"glue:GetSchema",
"glue:GetSchemaByDefinition",
"glue:GetSchemaVersion",
"glue:GetSchemaVersionsDiff",
"glue:GetSecurityConfiguration",
"glue:GetSecurityConfigurations",
"glue:GetTable",
"glue:GetTables",
"glue:GetTableVersion",
"glue:GetTableVersions"
"glue:GetTags",
"glue:GetTrigger",
"glue:GetTriggers",
"glue:GetUserDefinedFunction",
"glue:GetUserDefinedFunctions",
"glue:GetWorkflow",
"glue:GetWorkflowRun",
"glue:GetWorkflowRunProperties",
"glue:GetWorkflowRuns",
"glue:ListCrawlers",
"glue:ListDevEndpoints",
"glue:ListJobs",
"glue:ListMLTransforms",
"glue:ListRegistries",
"glue:ListSchemas",
"glue:ListSchemaVersions",
"glue:ListTriggers",
"glue:ListWorkflows",
"glue:QuerySchemaVersionMetadata",
"greengrass:DescribeComponent",
"greengrass:Get**",
"greengrass:List**",
"groundstation:DescribeContact",
"groundstation:GetConfig",
"groundstation:GetDataflowEndpointGroup",
"groundstation:GetMinuteUsage",
"groundstation:GetMissionProfile",
"groundstation:GetSatellite",
"groundstation:ListConfigs",
"groundstation:ListContacts",
"groundstation:ListDataflowEndpointGroups",
"groundstation:ListGroundStations",
"groundstation:ListMissionProfiles",
"groundstation:ListSatellites",
"groundstation:ListTagsForResource",
"guardduty:DescribeOrganizationConfiguration",
"guardduty:DescribePublishingDestination",
"guardduty:Get**",
"guardduty:List**",
"health:Describe**",
"iam:Generate**",
"iam:Get**",
"iam:List**",
"iam:Simulate**",
"imagebuilder:Get**",
"imagebuilder:List**",
"importexport:Get**",
"importexport:List**",
"inspector:Describe**",
"inspector:Get**",
"inspector:List**",
"inspector:Preview**",
"iot:Describe**",
"iot:Get**",
"iot:List**",
"iot1click:DescribeDevice",
"iot1click:DescribePlacement",
"iot1click:DescribeProject",
"iot1click:GetDeviceMethods",
"iot1click:GetDevicesInPlacement",
"iot1click:ListDeviceEvents",
"iot1click:ListDevices",
"iot1click:ListPlacements",
"iot1click:ListProjects",
"iot1click:ListTagsForResource",
"iotanalytics:Describe**"
"kendra:ListQuerySuggestionsBlockLists",
"kendra:ListTagsForResource",
"kendra:ListThesauri",
"kendra:Query",
"Kinesis:Describe***",
"Kinesis:Get***",
"Kinesis:List***",
"kinesisanalytics:Describe***",
"kinesisanalytics:Discover***",
"kinesisanalytics:Get***",
"kinesisanalytics:List***",
"kinesisvideo:Describe***",
"kinesisvideo:Get***",
"kinesisvideo:List***",
"Kms:Describe***",
"Kms:Get***",
"Kms:List***",
"lambda:Get***",
"lambda:List***",
"lex:DescribeBot***",
"lex:DescribeBotAlias***",
"lex:DescribeBotChannel***",
"lex:DescribeBotLocale***",
"lex:DescribeBotVersion***",
"lex:DescribeExport***",
"lex:DescribeImport***",
"lex:DescribeIntent***",
"lex:DescribeResourcePolicy***",
"lex:DescribeSlot***",
"lex:DescribeSlotType***",
"lex:Get***",
"lex:ListBotAliases***",
"lex:ListBotChannels***",
"lex:ListBotLocales***",
"lex:ListBots***",
"lex:ListBotVersions***",
"lex:ListBuiltInIntents***",
"lex:ListBuiltInSlotTypes***",
"lex:ListExports***",
"lex:ListImports***",
"lex:ListIntents***",
"lex:ListSlots***",
"lex:ListSlotTypes***",
"lex:ListTagsForResource***",
"license-manager:Get***",
"license-manager:List***",
"lightsail:GetActiveNames***",
"lightsail:GetAlarms***",
"lightsail:GetAutoSnapshots***",
"lightsail:GetBlueprints***",
"lightsail:GetBucketAccessKeys***",
"lightsail:GetBucketBundles***",
"lightsail:GetBucketMetricData***",
"lightsail:GetBuckets***",
"lightsail:GetBundles***",
"lightsail:GetCertificates***",
"lightsail:GetCloudFormationStackRecords***",
"lightsail:GetContainerAPIIMetadata***",
"lightsail:GetContainerImages***",
"lightsail:GetContainerServiceDeployments***",
"lightsail:GetContainerServiceMetricData***",
"lightsail:GetContainerServicePowers***",
"lightsail:GetContainerServices***",
"lightsail:GetDisk***",
"lightsail:GetDisks***",
"lightsail:GetDiskSnapshot***", 86
"lightsail:GetDiskSnapshots",
"lightsail:GetDistributionBundles",
"lightsail:GetDistributionLatestCacheReset",
"lightsail:GetDistributionMetricData",
"lightsail:GetDistributions",
"lightsail:GetDomain",
"lightsail:GetDomains",
"lightsail:GetExportSnapshotRecords",
"lightsail:GetInstance",
"lightsail:GetInstanceMetricData",
"lightsail:GetInstancePortStates",
"lightsail:GetInstances",
"lightsail:GetInstanceSnapshot",
"lightsail:GetInstancesSnapshots",
"lightsail:GetInstanceState",
"lightsail:GetKeyPair",
"lightsail:GetKeyPairs",
"lightsail:GetLoadBalancer",
"lightsail:GetLoadBalancerMetricData",
"lightsail:GetLoadBalancers",
"lightsail:GetLoadBalancerTlsCertificates",
"lightsail:GetOperation",
"lightsail:GetOperations",
"lightsail:GetOperationsForResource",
"lightsail:GetRegions",
"lightsail:GetRelationalDatabase",
"lightsail:GetRelationalDatabaseBlueprints",
"lightsail:GetRelationalDatabaseBundles",
"lightsail:GetRelationalDatabaseEvents",
"lightsail:GetRelationalDatabaseLogEvents",
"lightsail:GetRelationalDatabaseLogStreams",
"lightsail:GetRelationalDatabaseMetricData",
"lightsail:GetRelationalDatabaseParameters",
"lightsail:GetRelationalDatabases",
"lightsail:GetRelationalDatabaseSnapshot",
"lightsail:GetRelationalDatabaseSnapshots",
"lightsail:GetStaticIp",
"lightsail:GetStaticIps",
"lightsail:Is*",
"logs:Describe*",
"logs:FilterLogEvents",
"logs:Get*",
"logs:ListTagsLogGroup",
"logs:StartQuery",
"logs:StopQuery",
"logs:TestMetricFilter",
"lookoutvision:DescribeDataset",
"lookoutvision:DescribeModel",
"lookoutvision:DescribeProject",
"lookoutvision:ListDatasetEntries",
"lookoutvision:ListModels",
"lookoutvision:ListProjects",
"machinelearning:Describe*",
"machinelearning:Get*",
"macie:ListMemberAccounts",
"macie:ListS3Resources",
"macie2:BatchGetCustomDataIdentifiers",
"macie2:DescribeBuckets",
"macie2:DescribeClassificationJob",
"macie2:DescribeOrganizationConfiguration",
"macie2:GetAdministratorAccount",
"macie2:GetBucketStatistics",
"macie2:GetClassificationExportConfiguration",
"macie2:GetCustomDataIdentifier",
"macie2:GetFindings"
"macie2:GetFindingsFilter",
"macie2:GetFindingsPublicationConfiguration",
"macie2:GetFindingStatistics",
"macie2:GetInvitationsCount",
"macie2:GetMacieSession",
"macie2:GetMember",
"macie2:GetUsageStatistics",
"macie2:GetUsageTotals",
"macie2:ListClassificationJobs",
"macie2:ListCustomDataIdentifiers",
"macie2:ListFindings",
"macie2:ListFindingsFilters",
"macie2:ListInvitations",
"macie2:ListMembers",
"macie2:ListOrganizationAdminAccounts",
"macie2:ListTagsForResource",
"macie2:SearchResources",
"managedblockchain:GetNetwork",
"managedblockchain:GetProposal",
"managedblockchain:GetMember",
"managedblockchain:GetNode",
"managedblockchain:ListNetworks",
"managedblockchain:ListProposals",
"managedblockchain:ListProposalVotes",
"managedblockchain:ListInvitations",
"managedblockchain:ListMembers",
"managedblockchain:ListNodes",
"managedblockchain:ListTagsForResource",
"mediaconnect:DescribeFlow",
"mediaconnect:DescribeOffering",
"mediaconnect:DescribeReservation",
"mediaconnect:ListFlows",
"mediaconvert:DescribeEndpoints",
"mediaconvert:Get**",
"mediaconvert:List**",
"mediapackage:Describe**",
"mediapackage:List**",
"mediastore:DescribeContainer",
"mediastore:DescribeObject",
"mediastore:GetContainerPolicy",
"mediastore:GetCorsPolicy",
"mediastore:GetLifecyclePolicy",
"mediastore:GetMetricPolicy",
"mediastore:GetObject",
"mediastore:ListContainers",
"mediastore:ListItems",
"mediastore:ListTagsForResource",
"mgh:Describe**",
"mgh:GetHomeRegion",
"mgh:List**",
"mgn:DescribeJobLogItems",
"mgn:DescribeJobs",
"mgn:DescribeReplicationConfigurationTemplates",
"mgn:DescribeSourceServers",
"mgn:GetLaunchConfiguration",
"mgn:GetReplicationConfiguration",
"mobileanalytics:Get**",
"mobilehub:Describe**",
"mobilehub:Export**",
"mobilehub:Generate**",
"mobilehub:Get**",
"mobilehub:List**",
"mobilehub:Validate**",
"mobilehub:Verify**",
"mobiletargeting:Get**",
"mobiletargeting:List**", 

88
"monitron:GetProject",
"monitron:GetProjectAdminUser",
"monitron:ListProjects",
"monitron:ListTagsForResource",
"mq:Describe**",
"mq:List**",
"network-firewall:DescribeFirewall",
"network-firewall:DescribeFirewallPolicy",
"network-firewall:DescribeLoggingConfiguration",
"network-firewall:DescribeResourcePolicy",
"network-firewall:DescribeRuleGroup",
"network-firewall:ListFirewallPolicies",
"network-firewall:ListFirewalls",
"network-firewall:ListRuleGroups",
"network-firewall:ListTagsForResource",
"networkmanager:DescribeGlobalNetworks",
"networkmanager:GetConnections",
"networkmanager:GetCustomerGatewayAssociations",
"networkmanager:GetDevices",
"networkmanager:GetLinkAssociations",
"networkmanager:GetLinks",
"networkmanager:GetSites",
"networkmanager:GetTransitGatewayConnectPeerAssociations",
"networkmanager:GetTransitGatewayRegistrations",
"opsworks-cm:Describe**",
"opsworks-cm:List**",
"opsworks-cm:Describe**",
"opsworks:Get**",
"organizations:Describe**",
"organizations:List**",
"outposts:Get**",
"outposts:List**",
"personalize:Describe**",
"personalize:Get**",
"personalize:List**",
"pi:DescribeDimensionKeys",
"pi:GetDimensionKeyDetails",
"pi:GetResourceMetrics",
"polly:Describe**",
"polly:Get**",
"polly:List**",
"polly:SynthesizeSpeech",
"proton:GetEnvironment",
"proton:GetEnvironmentTemplate",
"proton:GetEnvironmentTemplateVersion",
"proton:GetService",
"proton:GetServiceInstance",
"proton:GetServiceTemplate",
"proton:GetServiceTemplateVersion",
"proton:GetEnvironmentAccountConnections",
"proton:GetEnvironments",
"proton:GetEnvironmentTemplates",
"proton:GetServiceInstances",
"proton:GetServices",
"proton:GetServiceTemplates",
"proton:GetTagsForResource",
"qldb:DescribeJournalS3Export",
"qldb:DescribeLedger",
"qldb:GetBlock",
"qldb:GetDigest",
"qldb:GetRevision",
"qldb:ListJournalS3Exports",
"qldb:ListJournalS3ExportsForLedger",
"qldb:ListLedgers",
"qldb:ListTagsForResource",
"ram:Get**", 
89
"ram:List*",
"rds:Describe*",
"rds:Download*",
"rds:List*",
"redshift:Describe*",
"redshift:GetReservedNodeExchangeOfferings",
"redshift:View*",
"rekognition:CompareFaces",
"rekognition:Detect*",
"rekognition:List*",
"rekognition:Search*",
"resource-groups:Get*",
"resource-groups:List*",
"resource-groups:Search*",
"robomaker:BatchDescribe*",
"robomaker:Describe*",
"robomaker:Get*",
"robomaker:List*",
"route53-recovery-cluster:Get*",
"route53-recovery-control-config:Describe*",
"route53-recovery-control-config:List*",
"route53-recovery-readiness:Get*",
"route53-recovery-readiness:List*",
"route53:Get*",
"route53:List*",
"route53:Test*",
"route53domains:Check*",
"route53domains:Get*",
"route53domains:List*",
"route53domains:View*",
"route53resolver:Get*",
"route53resolver:List*",
"s3-object-lambda:GetObject",
"s3-object-lambda:GetObjectAcl",
"s3-object-lambda:GetObjectRetention",
"s3-object-lambda:GetObjectTagging",
"s3-object-lambda:GetObjectVersion",
"s3-object-lambda:GetObjectVersionAcl",
"s3-object-lambda:GetObjectVersionTagging",
"s3-object-lambda:ListBucket",
"s3-object-lambda:ListBucketMultipartUploads",
"s3-object-lambda:ListBucketVersions",
"s3-object-lambda:ListMultipartUploadParts",
"s3:DescribeJob",
"s3:Get*",
"s3:List*",
"sagemaker:Describe*",
"sagemaker:GetSearchSuggestions",
"sagemaker:List*",
"sagemaker:Search",
"savingsplans:DescribeSavingsPlanRates",
"savingsplans:DescribeSavingsPlans",
"savingsplans:DescribeSavingsPlansOfferingRates",
"savingsplans:DescribeSavingsPlansOfferings",
"savingsplans:ListTagsForResource",
"schemas:Describe*",
"schemas:Get*",
"schemas:List*",
"schemas:Search*",
"sdb:Get*",
"sdb:List*",
"sdb:Select*",
"secretsmanager:Describe*",
"secretsmanager:GetResourcePolicy",
"secretsmanager:List*"
"securityhub:Describe*",
"securityhub:Get*",
"securityhub:List*",
"serverlessrepo:Get*",
"serverlessrepo:List*",
"serverlessrepo:SearchApplications",
"servicecatalog:Describe*",
"servicecatalog:GetApplication",
"servicecatalog:GetAttributeGroup",
"servicecatalog:List*",
"servicecatalog:Scan*",
"servicecatalog:Search*",
"servicediscovery:Get*",
"servicediscovery:List*",
"servicequotas:GetAssociationForServiceQuotaTemplate",
"servicequotas:GetAWSDefaultServiceQuota",
"servicequotas:GetRequestedServiceQuotaChange",
"servicequotas:GetServiceQuota",
"servicequotas:GetServiceQuotaIncreaseRequestFromTemplate",
"servicequotas:ListAWSDefaultServiceQuotas",
"servicequotas:ListRequestedServiceQuotaChangeHistory",
"servicequotas:ListRequestedServiceQuotaChangeHistoryByQuota",
"servicequotas:ListServiceQuotaIncreaseRequestsInTemplate",
"servicequotas:ListServiceQuotas",
"servicequotas:ListServices",
"ses:Describe*",
"ses:Get*",
"ses:List*",
"shield:Describe*",
"shield:Get*",
"shield:List*",
"signer:DescribeSigningJob",
"signer:GetSigningPlatform",
"signer:GetSigningProfile",
"signer:ListProfilePermissions",
"signer:ListSigningJobs",
"signer:ListSigningPlatforms",
"signer:ListSigningProfiles",
"signer:ListTagsForResource",
"snowball:Describe*",
"snowball:Get*",
"snowball:List*",
"sns:Check*",
"sns:Get*",
"sns:List*",
"sqs:Get*",
"sqs:List*",
"sqs:Receive*",
"ssm-contacts:DescribeEngagement",
"ssm-contacts:DescribePage",
"ssm-contacts:GetContact",
"ssm-contacts:GetContactChannel",
"ssm-contacts:ListContactChannels",
"ssm-contacts:ListContacts",
"ssm-contacts:ListEngagements",
"ssm-contacts:ListPageReceipts",
"ssm-contacts:ListPagesByContact",
"ssm-contacts:ListPagesByEngagement",
"ssm-incidents:GetIncidentRecord",
"ssm-incidents:GetReplicationSet",
"ssm-incidents:GetResourcePolicies",
"ssm-incidents:GetResponsePlan",
"ssm-incidents:GetTimelineEvent",
"ssm-incidents:ListIncidentRecords",
"ssm-incidents:ListIncidentRecords"
"ssm-incidents:ListResponsePlans",
"ssm-incidents:ListTagsForResource",
"ssm-incidents:ListTimelineEvents",
"ssm:Describe***",
"ssm:Get***",
"ssm:List***",
"sso-directory:Describe***",
"sso-directory:List***",
"sso-directory:Search***",
"sso:Describe***",
"sso:Get***",
"sso:List***",
"sso:Search***",
"states:Describe***",
"states:GetExecutionHistory",
"states:List***",
"storagegateway:Describe***",
"storagegateway:List***",
"sts:GetAccessKeyId***",
"sts:GetCallerIdentity",
"sts:GetSessionToken",
"support:DescribeCases",
"swf:Count***",
"swf:Describe***",
"swf:Get***",
"swf:List***",
"synthetics:Describe***",
"synthetics:Get***",
"synthetics:List***",
"tag:Get***",
"timestream:DescribeDatabase",
"timestream:DescribeEndpoints",
"timestream:DescribeTable",
"timestream:ListDatabases",
"timestream:ListMeasures",
"timestream:ListTables",
"timestream:ListTagsForResource",
"transcribe:Get***",
"transcribe:List***",
"transfer:Describe***",
"transfer:List***",
"transfer:TestIdentityProvider",
"trustedadvisor:Describe***",
"waf-regional:Get***",
"waf-regional:List***",
"waf:Get***",
"waf:List***",
"wafv2:CheckCapacity",
"wafv2:Describe***",
"wafv2:Get***",
"wafv2:List***",
"workdocs:CheckAlias",
"workdocs:Describe***",
"workdocs:Get***",
"worklink:Describe***",
"worklink:List***",
"workmail:Describe***",
"workmail:Get***",
"workmail:List***",
"workmail:Search***",
"workspaces:Describe***",
"xray:BatchGet***",
"xray:Get***
}

"Resource": "*
}
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, and links you to the AWS Config console, where you can view details about those external rules.

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"
    }
  ]
}
```
Programmatic roles and trust relationships for the AWS Control Tower audit account

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:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "config:PutConfigurationAggregator",
        "config:DeleteConfigurationAggregator",
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam:::role/service-role/AWSControlTowerConfigAggregatorRoleForOrganizations",
        "arn:aws:config::config-aggregator/"
      ],
    },
    {
      "Effect": "Allow",
      "Action": "organizations:EnableAWSServiceAccess",
      "Resource": "*"
    }
  ]
}
```

New resources created: `AWSControlTowerConfigAggregatorRoleForOrganizations` and `aws-controltower-ConfigAggregatorForOrganizations`.

When you are ready, you can enroll accounts individually, or enroll them as a group by registering an OU. When you’ve enrolled an account, if you create a rule in AWS Config, AWS Control Tower detects the new rule. The aggregator shows the number of external rules and provides a link to the AWS Config console where you can view the details of each external rule for your account. Use the information in the AWS Config console and the AWS Control Tower console to determine whether you have the appropriate guardrails enabled for the account.

**Note**

To link directly from the AWS Control Tower console to your aggregated list of AWS Config rules, configure your AWS Config console with the Config Recorder and Delivery Channel in the home Region of your management account.

---

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 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.

```
{
  "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.

```
{
  "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:

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

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.

{
  "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:
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 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
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSControlTowerAccountFactoryAccess",
            "Effect": "Allow",
            "Action": [
                "sso:GetProfile",
                "sso:CreateProfile",
                "sso:UpdateProfile",
                "sso:AssociateProfile",
                "sso:CreateApplicationInstance",
                "sso:GetSSOStatus",
            ],
            "Resource": [
                "arn:aws:iam::*:role/aws-controltower-ReadOnlyExecutionRole"
            ]
        }
    ]
}
```
"sso:GetTrust",
"sso:CreateTrust",
"sso:UpdateTrust",
"sso:GetPeregrineStatus",
"sso:GetApplicationInstance",
"sso:ListDirectoryAssociations",
"sso:ListPermissionSets",
"sso:GetPermissionSet",
"sso:ProvisionApplicationInstanceForAWSAccount",
"sso:ProvisionApplicationProfileForAWSAccountInstance",
"sso:ProvisionSAMLProvider",
"sso:ListProfileAssociations",
"sso-directory:ListMembersInGroup",
"sso-directory:AddMemberToGroup",
"sso-directory:SearchGroups",
"sso-directory:SearchGroupsWithGroupName",
"sso-directory:SearchUsers",
"sso-directory:CreateUser",
"sso-directory:DescribeGroups",
"sso-directory:DescribeDirectory",
"sso-directory:GetUserPoolInfo",
"controltower:CreateManagedAccount",
"controltower:DescribeManagedAccount",
"controltower:DeregisterManagedAccount",
"s3:GetObject",
"organizations:describeOrganization",
"sso:DescribeRegisteredRegions"
],
"Resource": "*"}
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) Region
- Asia Pacific (Sydney)
- Asia Pacific (Singapore) Region
- Europe (Frankfurt) Region
- Europe (Ireland)
- Europe (London) Region
- Europe (Stockholm) Region
- Asia Pacific (Mumbai) Region
- Asia Pacific (Seoul) Region
- Asia Pacific (Tokyo) Region
- Europe (Paris) Region
- South America (São Paulo) Region

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.

Guardrails and Regions

Currently, all preventive guardrails work globally. Detective guardrails, however, only work in Regions where AWS Control Tower is supported. For more information about the behavior of guardrails when you activate AWS Control Tower in a new Region, see Configure your AWS Control Tower Regions (p. 99).

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 guardrails 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 guardrails 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 guardrails on those OUs and accounts.
- **Your existing detective guardrails 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 guardrails that already are enabled on the OU will begin working on that account in the newly configured Regions.

### Configure AWS Control Tower Regions

2. In the left-pane navigation menu, choose Landing zone settings.
3. On the Landing zone settings page, in the Details section, choose the Modify settings button in the upper right. You are directed to the update landing zone workflow, because governing new Regions, or removing Regions from governance, requires you to update to the latest landing zone version.
4. Under Additional AWS Regions for governance, search for the Regions you want to govern (or stop governing). The State column indicates which Regions you currently govern, and which ones you don’t.
5. Select the checkbox for each additional Region to govern. Deselect the checkbox for each Region from which you are removing governance.

   **Note**

   If you opt not to govern a Region, you can still deploy resources in that Region, but those resources will remain outside of AWS Control Tower governance.

6. Complete the rest of the workflow, then choose Update landing zone.
7. When the landing zone setup completes, Re-register the OUs to update the accounts in your new Regions. For more information, see Update existing OUs and accounts (p. 169).
An alternative method of provisioning or updating individual accounts after configuring new Regions is by using the API framework of AWS Service Catalog and the AWS CLI to update the accounts in a batch process. For more information, see Provisioning and updating accounts using script automation (p. 37).

Configure the Region deny guardrail

The Region deny guardrail is unique, because it applies to the landing zone as a whole, rather than to any specific OU. To configure the Region deny guardrail, go to the Landing zone settings page and select Modify settings.

- This setting can be changed at a later time.
- When enabled, this guardrail applies to all registered OUs.
- This guardrail cannot be configured for individual OUs.

**Note**
Before you enable the Region deny guardrail, 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 guardrail. While the guardrail is enabled, you will not be able to deploy resources in the denied Regions.

The Region deny guardrail 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 guardrail 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 guardrail. To learn more, see Deny access to AWS based on the requested AWS Region (p. 210).

When you enable the guardrail, 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 guardrail, 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 guardrail 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 guardrail with preventive guidance.

To view the template for the Region deny guardrail SCP, see Deny access to AWS based on the requested AWS Region (p. 210) 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. 103). For information about enrolling multiple accounts into AWS Control Tower, see Register an existing organizational unit with AWS Control Tower (p. 167).

Methods of provisioning

AWS Control Tower provides four methods for creating and updating member accounts. Two methods are primarily console-based, and two 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 AWS Service Catalog. If your landing zone is not in a state of drift, you can use Enroll account as a method to add new accounts from the console, as well as to enroll existing AWS accounts into AWS Control Tower.

If you intend to set up or manage many accounts as a group, you may prefer an automated method: either to configure new accounts programmatically using IAM roles and Lambda functions, or to use a GitOps approach with AFT.

Console-based methods:

- Through the Account Factory console that is part of AWS Service Catalog. See Provision and manage accounts with Account Factory (p. 117).
- Through the Enroll account feature within AWS Control Tower, if your landing zone is not in a state of drift. See Create or Enroll An Individual Account (p. 118).

Automated methods:

- 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. 97).
- From the AWS Control Tower Account Factory for Terraform (AFT), which relies on Account Factory and a GitOps model to allow automation of account provisioning and updating. See Provision accounts with AWS Control Tower Account Factory for Terraform (p. 124).

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 Create or Enroll An Individual Account (p. 118).
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 AWS Service Catalog console, or by calling the AWS Service Catalog `ProvisionProduct` API.
2. AWS 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 guardrails.
5. AWS 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, AWS 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.

For general information about permissions required in AWS Control Tower, see Using Identity-Based Policies (IAM Policies) for AWS Control Tower (p. 240). For information about roles and accounts in AWS Control Tower, see How AWS Control Tower works with roles to create and manage accounts (p. 73).

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.

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

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 (p. 235).

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, AWS Service Catalog provisioned products, and SSO users. AWS Control Tower also deploys resources, as required by the guardrail configuration, for the 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. 29).
View your accounts

The **Accounts** page lists all 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.

On the **Accounts** 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 guardrails configured for its registered parent OU, but the OU's detective guardrails do not apply to this account. If the parent OU is unregistered, no guardrails apply to this account.
- **Enrolling** – Your account is being brought into governance by AWS Control Tower. We are aligning the account with the guardrail configuration for the parent OU. This process may require several minutes per account resource.
- **Enrolled** – The account is governed by the guardrails configured for its parent OU. It is fully managed by AWS Control Tower.
- **Enrollment failed** – AWS Control Tower could not enroll the account. For more information, see [Common causes for failure of enrollment](#) (p. 108).

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 your audit and log archive accounts when you're setting up your landing zone, but you cannot change the names after setup. (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)

Management account

This AWS account launches AWS Control Tower. By default, the root user for this account and the 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 AWS SSO user with Administrator privileges when performing administrative functions within the AWS Control Tower management console, instead of the signing in as the root or IAM administrator users 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

This shared account is created automatically when you create your landing zone. Note that the email address must not already have an associated AWS account.

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 created automatically when you create your landing zone. Note that the email address must not already have an associated AWS account.

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 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 (p. 148).

Audit notifications that are triggered within a managed 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 managed 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. 7) Also see Programmatic roles and trust relationships for the AWS Control Tower audit account (p. 94).

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 SSO users with Admin privileges in the AWS 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 (p. 117)
- Automating tasks in AWS Control Tower (p. 45)
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.

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.

To learn more about trusted access and AWS CloudFormation StackSets, see AWS CloudFormation StackSets and AWS Organizations. 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.

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-CLUSTRAIL
  - AWSControlTowerBP-BASELINE-CLOUDWATCH
  - AWSControlTowerBP-BASELINE-CONFIG
  - AWSControlTowerBP-BASELINE-ROLES
  - AWSControlTowerBP-BASELINE-SERVICE-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 Single Sign-On 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 guardrails to the account by means of the SCPs that apply to the selected OU as a whole.

- Adds the AWS Config rules that apply the AWS Control Tower detective guardrails to the account.

**Note**

When you enroll the account into AWS Control Tower, your account is governed by the AWS CloudTrail trail for the new 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.
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.

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 guardrails. 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.

Prerequisites for Enrollment

These prerequisites are required before you can enroll an account in AWS Control Tower:

1. We recommend that the account should not have an AWS Config configuration recorder or delivery channel. These may be deleted through the AWS CLI before you can enroll an account. Otherwise, see [Enroll accounts that have existing AWS Config resources](#) for instructions on how you can modify your existing resources.

2. 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.

3. Before you can enroll an existing account in AWS Control Tower, the account must have the following roles, permissions, and trust relationships in place. Otherwise, enrollment will fail.

   **Role Name:** AWSControlTowerExecution

   **Role Permission:** AdministratorAccess (AWS managed policy)

   **Role Trust Relationship:**

   ```json
   {
       "Version": "2012-10-17",
       "Statement": [
         {
           "Effect": "Allow",
           "Principal": {
   ```
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 account**

After the AdministratorAccess permission 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 Account Factory page and select Enroll account.
- Specify the current email address of the existing account you'd like to enroll in AWS Control Tower.
- Specify the first and last name of the account owner.
- Specify the organizational unit (OU) in which you'd like to enroll the account.
- Choose Enroll account.

**Common causes for failure of enrollment**

- Your IAM principal may lack the necessary permissions to provision an account. To enroll an existing account, the AWSControlTowerExecution role must be present in the account you’re enrolling.
- 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 must not have a configuration recorder or delivery channel, so these must be deleted through the AWS CLI before you can enroll an account. For more information, see Interacting with AWS Control Tower using AWS CloudShell (p. 42).
- 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.
What if the account does not meet the prerequisites?

To fulfill the prerequisites 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 into the AWS Control Tower organization. You can follow this procedure given in the AWS Organizations documentation, Inviting an AWS account to join your organization.
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. It establishes SCPs and consolidated billing.
4. Now you must fulfill the remaining enrollment prerequisites:
   - Create the necessary role.
   - Clear out the default VPC. (This part is optional—AWS Control Tower does not change your existing default VPC.)
   - Delete or modify any existing AWS Config configuration recorder or delivery channel through the AWS CLI or AWS CloudShell.
   - Any other prerequisites, as needed.
5. Enroll the account into AWS Control Tower. This step brings the account into full AWS Control Tower governance.

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`

**Note**

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 drops out of its existing organization.
Optional steps for deprovisioning an account so it can be enrolled, keeping its stack

1. Optionally, to keep the applied CFN, delete the stack instance from the stack sets, making sure to 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 does not actually delete the account.)
3. Optionally, set up the account with the necessary billing details, as required for any account that does not belong to an organization, then remove the account from the organization. You would do this so that the account does not count against the total in your AWS Organizations quota.
4. Clean up the account, if resources remain, and close it, following account closure steps given in **Unmanage a member account** (p. 122).
5. If you have a **Suspended** OU with defined guardrails, you can move the account there instead of doing Step 1.

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. 107) 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 into your organization's management account.
2. Go to **AWS Organizations**.
3. Under **Accounts**, select the account you want to enroll and copy its account ID.
4. Open the account dropdown menu on the top navigation bar and choose **Switch Role**.
5. On the **Switch role** form, fill in the following fields:
   - Under **Account**, enter the account ID you copied.
   - Under **Role**, enter the name of the IAM role that enables cross-account access to this account. The name of this role was defined when the account was created. If you did not specify a role name when you created the account, enter the default role name, **OrganizationAccountAccessRole**.
6. Choose **Switch Role**.
7. You should now be signed into the AWS management console as the child account.
8. When you're finished, stay in the child account for the next part of the procedure.
9. Make note of the management account ID, because you will need to enter it in the next step.

Step 2: Give AWS Control Tower permission to manage the account.

1. Go to IAM.
2. Go to Roles.
3. Choose Create role.
4. 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.
5. When asked to attach policies, choose AdministratorAccess.
6. Choose Next:Tags.
7. You may see an optional screen titled Add tags. Skip this screen for now by choosing Next:Review.
8. On the Review screen, in the Role name field, enter AWSControlTowerExecution.
9. Enter a brief description in the Description box, such as Allows full account access for enrollment.
10. Choose Create role.
11. Navigate to the role you just created. Choose Roles on the left. Select AWSControlTowerExecution.
13. 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",
   "Principal": {
   "AWS": "arn:aws:iam::Management Account ID:root"
   },
   "Action": "sts:AssumeRole",
   "Condition": {}  
   }
   ]
}
```

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 Account Factory page in AWS Control Tower, choose Enroll account. Fill in the required fields. Use the email address associated with the account you just updated.
   - Specify the current email address of the existing account you'd like to enroll in AWS Control Tower.
   - Specify the first and last name of the account owner.
   - Specify the organizational unit (OU) in which you'd like to enroll the account.
1. **Choose Enroll account.**

2. **Verify enrollment.**
   - From AWS Control Tower, choose **Accounts**.
   - 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.

```yaml
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:
          Version: 2012-10-17
          Statement:
            - Effect: Allow
              Principal:
                AWS:
                  - !Ref AdministratorAccountId
              Action:
                - sts:AssumeRole
              Path: /
              ManagedPolicyArns:
                - arn:aws:iam::aws:policy/AdministratorAccess
```

**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.
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

- 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.

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. 113), 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.

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 guardrails 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

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

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/AWSConfigRole
        - arn:aws:iam::aws:policy/ReadOnlyAccess

3. Provide the name for the stack as CustomerCreatedConfigRecorderRoleForControlTower
4. Create the stack.

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.
Step 5a. AWS Config recorder resources

Only one AWS Config recorder can exist per AWS Region. If another exists, modify the settings as shown.

- **Name**: DON’T CHANGE
- **RoleARN**: IAM_ROLE_ARN
  - **RecordingGroup**: AllSupported: true
  - **IncludeGlobalResourceTypes**: true
  - **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=true --region CURRENT_REGION
```

Step 5b. Modify AWS Config delivery channel resources

Only one AWS Config delivery channel can exist 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-HOME_REGION
  ```
- **S3KeyPrefix**: ORGANIZATION_ID
- **SnsTopicARN**: The SNS topic ARN from the audit account, with the following format:

  ```
  arn:aws:sns:CURRENT_REGION:AUDIT_ACCOUNT:aws-controltower-
  AllConfigNotifications
  ```

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.
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:

```
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

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: true
        ResourceTypes: []
  CustomerCreatedConfigDeliveryChannel:
    Type: AWS::Config::DeliveryChannel
    Properties:
      Name: aws-controltower-BaselineConfigDeliveryChannel-customer-created
      ConfigSnapshotDeliveryProperties:
        DeliveryFrequency: TwentyFour_Hours
        S3BucketName: aws-controltower-logs-LOGGING_ACCOUNT_ID-HOME_REGION
        S3KeyPrefix: ORGANIZATION_ID
```

116
CustomerCreatedAggregationAuthorization:
  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 AuthorizedAccountId 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 your AWS Control Tower landing zone with Account Factory.

**Permissions for Configuring and Provisioning Accounts**

The AWS Control Tower account factory enables cloud administrators and AWS Single Sign-On end users to provision accounts in your landing zone. By default, AWS SSO 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 generous permissions across your organization.

The AWS Control Tower management account has a trust relationship with the AWSControlTowerExecution role, which enables 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. 73).

*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. 106).
Create or Enroll An Individual Account

The Enroll account feature is available in the AWS Control Tower console, for provisioning new accounts in your landing zone and for enrolling existing AWS accounts so that they are governed by AWS Control Tower.

The Enroll account capability is available when your landing zone is not in a state of drift. To view this capability:

- Navigate to the Account Factory page in AWS Control Tower.
- Select the Enroll account item near the top of the page.
- You'll then see a Create account section, where you can fill in the required fields: account email, account name, SSO user name, and organizational unit.
- When you've filled in the information, select Enroll account.

You'll see a flashbar confirming that your account enrollment process has been successfully submitted. If an error has occurred, AWS Control Tower may ask you for corrections. The account provisioning process may take several minutes.

Note
If you are enrolling an existing AWS account, be sure to type the existing email address correctly. Otherwise, a new account will 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 AWS Service Catalog 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 an IAM user that has the AWSServiceCatalogEndUserFullAccess policy enabled, along with permissions to use the AWS Control Tower console, and you cannot be signed in as Root.

Accounts that you enroll must 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 Service Catalog (p. 119).

Provision Account Factory accounts with AWS Service Catalog

The following procedure describes how to provision accounts as an AWS SSO end user, through AWS Service Catalog. This procedure also is referred to as advanced account provisioning. We recommend using the Enroll account capability whenever possible.

To provision accounts in Account Factory as an end 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 home region.
6. Search for and choose Service Catalog to open the AWS Service Catalog console.
7. From the navigation pane, choose **Products list**.

8. Select **AWS Control Tower Account Factory**, then choose the **Launch** button. This selection starts the wizard to provision a new account.

9. Fill in the information, and keep the following in mind:
   
   - The **SSOUserEmail** can be a new email address, or the email address associated with an existing AWS SSO 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 **SSOUserEmail**, you can use that email address here.

10. When you're finished, choose **Next** until you get to the **Review** page of the wizard. Do not define **TagOptions** and do not enable **Notifications**, otherwise the account can fail to be provisioned.

11. Review your account settings, and then choose **Launch**. Do not 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**
   
   Only one account can be provisioned at a time.

**Considerations on Managing Account Factory Accounts**

Accounts that you provision through the AWS Control Tower Account Factory can be updated, they can be closed, or they can be repurposed. For example, you can repurpose existing accounts for other workloads and other users by updating the user parameters for the account.

If you specify a new SSO 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 SSO user account. The previously created user account is not removed. If you prefer to remove the previous SSO user email from AWS SSO, see **Disabling a User**.

With Account Factory you also can change the organizational unit (OU) for an account, or you can unmanage an account, by following the procedures in this chapter. For more information on unmanaging an account, see **Unmanage a member account** (p. 122). Certain updates require that you or an administrator must **Sign in as a Root User** (p. 30) to the account, to gain appropriate permissions.

**Update and move account factory accounts with AWS Service Catalog**

The following procedure guides you through how to update your Account Factory account or move it to a new OU, through AWS Service Catalog, by updating the provisioned product.

**To update an Account Factory account or change its OU**

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 be signed in as a user with the permissions to provision new products in AWS Service Catalog; for example, an AWS SSO user in either the **AWSAccountFactory** or **AWSServiceCatalogAdmins** groups.

2. From the navigation pane, choose **Provisioned products list**.
3. For each account listed, perform the following steps to update all your member accounts:
   a. From the drop-down menu for the account, choose Provisioned product details.
   b. Make a note of the following parameters:
      - **SSOUserEmail** (Available in provisioned product details)
      - **AccountEmail** (Available in provisioned product details)
      - **SSOUserFirstName** (Available in SSO)
      - **SSOUserLastName** (Available in SSO)
      - **AccountName** (Available in SSO)
   c. From Actions, choose Update.
   d. Choose the button next to the Version of the product you want to update, and choose Next.
   e. 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.
   f. Choose Next.
   g. Review your changes, and then choose Update. This process can take a few minutes per account.

Change email address of an enrolled account

Follow the procedure given in this section to change the email address of an enrolled member account. This procedure does not allow you to change the email address of the Management account, Log Archive account, or Audit account.

**To change the email address 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.
3. Change the email address as you would for any other AWS account.
4. You may experience a delay before AWS Organizations reflects the update. Wait for the email address change to be reflected in AWS Organizations.
5. Update the provisioned product in AWS Service Catalog using the account's previous email address. The update process associates the new email address with the provisioned product, so that the email update takes effect in AWS Control Tower. Subsequent provisioned product updates are made with the new email address.

To change the password or email address of a member account you created with AWS Organizations, see Accessing a member account as the root user, in the AWS Organizations documentation.

**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.
Configure Amazon VPC settings

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.
5. AWS Control Tower automatically updates itself to reflect the name change. This update will not be reflected in the provisioned product in AWS Service Catalog.

Configure Account Factory with Amazon Virtual Private Cloud settings

Account Factory enables 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.

  **Note**
  
  If you set the account factory VPC configuration so that public subnets are enabled when provisioning a new account, account factory configures Amazon VPC to create a NAT Gateway. You will be billed for your usage by Amazon VPC. See VPC Pricing for more information.

- Choose the maximum number of private subnets in Amazon VPC from the list. By default, 1 is selected. The maximum number of private subnets allowed is 2 per availability zone.

- Enter the range of IP addresses for creating your account VPCs. The value must be in the form of a classless inter-domain routing (CIDR) block (for example, the default is 172.31.0.0/16). This CIDR block provides the overall range of subnet IP addresses for the VPC that Account Factory creates for your account. Within your VPC, subnets are assigned automatically from the range you specify, and they are equal in size. By default, subnets within your VPC do not overlap. However, subnet IP address ranges in the VPCs of all your provisioned accounts could overlap.

- Choose a region or all the regions for creating a VPC when an account is provisioned. By default all available regions are selected.
• From the list, choose the number of Availability Zones to configure subnets for in each VPC. The default and recommended number is 3.
• Choose Save.

You can set up these configuration options to create new accounts that don't include a VPC. See the walkthrough.

Unmanage a member account

If you created an account in Account Factory that you no longer want to be managed by AWS Control Tower in a landing zone, you can unmanage the account. This can be done in the AWS Service Catalog console by an AWS SSO user in the AWSAccountFactory group. For more information on AWS SSO users or groups, see Managing Users and Access Through AWS Single Sign-On (p. 229). The following procedure describes how to unmanage a member account.

To unmanage a member account

2. From 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 to no longer manage.
4. On the Provisioned product details page, from the Actions menu, choose Terminate.
5. From the dialog box that appears, choose Terminate.

   Important
   The word terminate is specific to AWS Service Catalog. When you terminate an Account Factory account in AWS Service Catalog, the account is not closed. This action removes the account from its OU and your landing zone.

6. When the account has been unmanaged, its status changes to Not Enrolled.
7. If you no longer need the account, close it. For information about closing AWS accounts, see Closing an Account in the AWS Billing User Guide

Note
An unmanaged account is not closed or deleted. When the account has been unmanaged, the AWS SSO 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 AWS SSO by updating the account in Account Factory and changing the AWS SSO user email address for the account. For more information, see Update and move account factory accounts with AWS Service Catalog (p. 119).

You can view an AWS YouTube video that explains how to remove and close down an account 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.
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>
<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-SecurityNotifications</td>
</tr>
<tr>
<td>Service</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWS Lambda</td>
<td>Applications</td>
<td>StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-*</td>
</tr>
</tbody>
</table>
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. You'll create an `account request` Terraform file, which provides the necessary input that triggers the AFT workflow for account provisioning.

When account provisioning or updating is complete, the AFT workflow continues. It runs the AFT account provisioning framework and account customizations steps.

**Note**
AFT does not affect workflow performance in AWS Control Tower. For example, if you provision an account through Account Factory, or through AFT, the same back-end workflow occurs, without any timing differences. AFT enables automation, therefore the time required for provisioning may be less noticeable.

### Prerequisites

Before AFT can begin to provision accounts, some prerequisite steps are required. You must create:

- **A fully deployed AFT environment.** For more information about AFT deployment, see one or more of these sections:
  - Deploy AWS Control Tower Account Factory for Terraform (AFT) (p. 126)
  - Overview of AWS Control Tower Account Factory for Terraform (p. 128)
  - Configure and launch your AWS Control Tower Account Factory for Terraform (p. 126)
- **One or more AFT git repositories in your environment.** For more information, see Post-deployment steps (p. 127) in Deploy AWS Control Tower Account Factory for Terraform (AFT) (p. 126).
- **An account template folder in your chosen git repository for account customizations.** This folder is optional.

### Provision a new account with AFT

To begin provisioning, set up a `create account request` file that contains appropriate input values for the following parameters in your chosen git repository. You can view an example account request file, which you can push to the git repository you've set up for storing your AFT account requests. For more information, see Step 3 of your post-deployment steps.

When you give the `git push` command, it triggers the `ct-aft-account-request` operation in the AWS CodePipeline in the AFT management account, to process your account request.

**Include the following list of parameters in your file:**

- The value of **module name** must be unique per AWS account request.
- The value of **module source** is the path to the account request Terraform module provided by AFT.
• The value of `control_tower_parameters` captures the input required to create an AWS Control Tower account, as follows.
  • `AccountEmail`
  • `AccountName`
  • `ManagedOrganizationalUnit`
  • `SSOUserEmail`
  • `SSOUserFirstName`
  • `SSOUserLastName`

For more information about setting up accounts in AWS Control Tower, see Provision and manage accounts with Account Factory.

• The parameter `account_tags` captures user-defined keys and values that can tag AWS accounts according to your business criteria. See the AWS Organizations documentation for more information on account tags.

• The parameter `change_management_parameters` captures additional information you may want to keep, such as the reason for the account request, and an identifier for who initiated the request.
  • `change_requested_by`
  • `change_reason`

• The parameter `custom_fields` captures custom keys and values. You may want to collect additional metadata that can be logged with the account request. This metadata can trigger additional processing, either during provisioning or when updating an account. You can refer to this metadata during account customizations to determine the proper guardrails to deploy. For example, an account that is subject to regulatory compliance could deploy an additional AWS Config rule.

• The parameter `account_customizations_name` is optional. It captures the specified account template folder for account customizations. See Account customizations (p. 141) for more information.

### Update an existing account

You may update existing accounts that AFT has provisioned by updating previously submitted account requests. The `git push` action triggers the account provisioning workflow, which processes an account update request.

When you're updating an existing account, AFT allows you to update the `ManagedOrganizationalUnit` field of the `control_tower_parameters` input, and other fields that are NOT part of the `control_tower_parameters` input. The input to `control_tower_parameters`, which you provided during provisioning, cannot be changed.

**Note**

You can update existing AWS Control Tower accounts that were created outside of AFT by specifying the account in the `aft-account-request` repository.

### Submit multiple account requests

Although AWS Control Tower account factory can process a single request at any time, the AFT pipeline allows you to submit multiple account requests. AFT queues all the requests to be processed by AWS Control Tower account factory, and they are processed in first-in, first-out (FIFO) order.

You can create a single account request Terraform file for each account, or you can cascade multiple requests in a single Terraform file.
Deploy AWS Control Tower Account Factory for Terraform (AFT)

This section is intended 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
AFT is deployed with a Terraform module, available in the AFT repository. We recommend that you refer to AFT Terraform modules from the GitHub source, and do not clone it, so that you can control and consume updates to the modules as they are available.

Deployment prerequisites

AFT has prerequisites for installation and for running. This guide assumes that you can create and interact with each of the following resources required to deploy the AFT solution:

- An AWS Control Tower landing zone.
- A Terraform version and distribution. See Terraform version supported (p. 130) for more details about compatible AFT Terraform distributions.
- A home Region for your AWS Control Tower landing zone.
- If you are not choosing AWS CodeCommit – you must set up available git-based repositories if you plan to use a third-party version control system (VCS). See Alternatives for version control of source code in AFT (p. 144) for instructions on how to set up these repositories in your environment.
- A runtime environment in which to run the Terraform module that installs AFT.
- AFT feature options that you can select as part of your AFT deployment. For more information, see Enable feature options (p. 132).

Note
The easiest deployment for AFT is to choose AWS CodeCommit as your VCS and Terraform OSS as your distribution. If you choose another VCS or Terraform distribution, be sure to read the customization instructions provided in other sections of this guide.

Configure and launch your AWS Control Tower Account Factory for Terraform

Five steps are required to configure and launch your AFT environment.

Step 1: Launch your AWS Control Tower landing zone

Before launching AFT, you must have a working AWS Control Tower landing zone in your AWS account. You will configure and launch AFT from the AWS Control Tower management account.

Step 2: Create a new organizational unit for AFT (recommended)

We recommend that you create a separate OU in your AWS Organization, where you will deploy the AFT management account. Create an OU through your AWS Control Tower management account. For instructions on how to create an OU, refer to Create an organization in the AWS Organizations User Guide.

Step 3: Provision the AFT management account

AFT requires a separate AWS account to manage and orchestrate its own requests. From the AWS Control Tower management account that's associated with your AWS Control Tower landing zone, you'll provision this account for AFT.
To provision the AFT management account, see Provision Account Factory accounts with AWS Service Catalog (p. 118). When specifying an OU, be sure to select the OU you created in Step 2. When specifying a name, use \textit{AFT-Management}.

\textbf{Note}

It can take up to 30 minutes for the account to be fully provisioned. Validate that you have access to the AFT management account.

**Step 4: Ensure that the Terraform environment is available for deployment**

This step assumes that you are experienced with Terraform, and that you have procedures in place for executing Terraform. AFT supports Terraform Version 0.15.x or later.

**Step 5: Call the Account Factory for Terraform module to deploy AFT**

The Account Factory for Terraform module must be called while you are authenticated with \textbf{AdministratorAccess} credentials in your AWS Control Tower management account.

AWS Control Tower, through the AWS Control Tower management account, vends a Terraform module that establishes all infrastructure necessary to orchestrate your AWS Control Tower account factory requests. You can view that module in the AFT repository. Refer to the module’s README file for information about the input required to run the module and deploy AFT.

If you have established pipelines for managing Terraform in your environment, you can integrate this module into your existing workflow. Otherwise, run the module from any environment that is authenticated with the required credentials.

\textbf{Note}

The AFT Terraform module does not manage a backend Terraform state. Be sure to preserve the Terraform state file that’s generated, after applying the module, or set up a Terraform backend using Amazon S3 and DynamoDB.

Certain input variables may contain sensitive values, such as a private \texttt{ssh} key or Terraform token. These values may be viewable as plain text in Terraform state file, depending on your deployment method. It is your responsibility to protect the Terraform state file, which may contain sensitive data. See the Terraform documentation for more information.

\textbf{Note}

Deploying AFT through the Terraform module requires several minutes. Initial deployment may require up to 30 minutes. As a best practice, use AWS Security Token Service (STS) credentials and ensure that the credentials have a timeout sufficient for a full deployment, because a timeout causes the deployment to fail. The minimum timeout for AWS STS credentials is 60 minutes or more. Alternatively, you can leverage any IAM user that has \textbf{AdministratorAccess} permissions in the AWS Control Tower management account.

**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. 144) 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) Grant access to AWS Control Tower account factory portfolio

AFT requires access to the AWS Control Tower account factory portfolio in AWS Service Catalog for the AFT management account, so AFT can process your account provisioning requests. To enable portfolio access, grant permission for the AWSAFTExecution role, for the AWS Control Tower account factory portfolio. See the AWS Service Catalog documentation for instructions on how to grant access to the AWS Service Catalog portfolios.

Step 3: (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. 124).
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. 140).
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. 142).
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. 142).

AFT expects each of these repositories to follow a specific directory structure. Templates with which to populate your repositories, and instructions for how to populate those templates, are available in the Account Factory for Terraform module in the AFT github repository.

Overview of AWS Control Tower Account Factory for Terraform

AWS Control Tower Account Factory for Terraform (AFT) sets up a Terraform pipeline that helps you provision and customize your accounts in AWS Control Tower. It is designed to provide the advantage of Terraform-based account provisioning, while maintaining AWS Control Tower governance of your accounts.

You’ll create an account request Terraform file, which provides the necessary input that triggers the AFT workflow for account provisioning. After provisioning is complete, AFT can run additional customization steps, automatically.

AFT supports Terraform Cloud, Terraform Enterprise, and Terraform Open Source. It provides the ability to initiate account creation with an input file and a simple git push command, as well as the ability to customize new or existing accounts. Account creation includes all AWS Control Tower governance benefits, and account customizations that assist you to meet your organization’s standard security procedures and compliance guidelines.

AFT combines the capabilities of other AWS services as Component services (p. 138), 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.

Video Walkthrough of Automated Account Provisioning in AWS Control Tower.

AFT Architecture

The AFT architecture has three high-level principles.

1. **AFT Pipeline** – (Mandatory) These components are necessary to support the AFT framework. They are part of every deployment, and supported by the AWS Control Tower team.

2. **AFT Feature Options** – (Optional) These components are embedded in the offering, and built according to AWS prescriptive guidance. These options are deployed with feature flags that you can select, and they are supported by the AWS Control Tower team.

   **Feature options include**
   - AWS CloudTrail data event logging for Amazon S3 buckets and Lambda; functions.
   - Automated deletion of the AWS default VPC for an account, in all AWS Regions.
   - Automatic enrollment in the AWS Enterprise Support plan.

3. **AFT Customizations** – (Optional) You can add any public or private Terraform module that you choose into the AFT pipeline. You are responsible for all module configurations related to these customizations.

AWS Control Tower Account Factory for Terraform – architecture diagram
Order of operations

You will run AFT operations while signed into the AFT management account. For a full account provisioning workflow, the order of stages, shown in the diagram from left to right, is as follows:

1. Each account is provisioned. This stage runs in the AFT management account.
2. If you've set up a state machine, certain customizations are run as a part of provisioning (referred to as the provisioning framework). This stage runs in the AFT management account.
3. Global customizations are run. This stage runs in the pipeline that's set up for each vended account.
4. Account customizations are run, if they were referenced in the initial account provisioning request. This stage runs on targeted accounts only.

If the account is already provisioned, you must initiate further customizations manually in that 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 version supported

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.

```text
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. 126).

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, and it initiates an 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 Terraform Cloud.

For selecting Terraform Cloud as your distribution, provide the following input parameters:

- `terraform_distribution = "tfc"`
- `terraform_token` – This parameter contains the value of your Terraform Cloud 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. 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.

See the Terraform documentation to learn more about how to set up Terraform Cloud.

**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.

**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. 124) 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. 133)
AWS Control Tower User Guide
Enable feature options

- AWS Enterprise Support plan (p. 133)
- Delete the AWS default VPC (p. 134)

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.

```bash
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.

```bash
aft_feature_enterprise_support=false
```

Refer to Compare AWS Support Plans to learn more about AWS Support Plans.
Delete the AWS default VPC

When this option is enabled, the AFT pipeline deletes the associated AWS default VPCs in all AWS Regions, for the accounts that AFT provisions.

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. AFT provides an automated way to delete the default VPCs for the accounts that AFT provisions.

To enable this option, set the following feature flag to True in your AFT deployment input configuration:

```plaintext
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:

  ```plaintext
  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>AWSAFTAdministratorRole</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AWSAFTExecutionRole</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>AWS service</td>
<td>Resource type</td>
<td>Resource name</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------</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, aws-ct-aft-request-audit, 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, aws-ct-aft-prebaseline-customizations, aws-ct-aft</td>
</tr>
<tr>
<td></td>
<td></td>
<td>trigger-baseline, 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, aws-ct-aft-failure-notifications</td>
</tr>
<tr>
<td>Amazon EventBridge</td>
<td>Event buses</td>
<td>aws-ct-aft-events-from-ct-management</td>
</tr>
<tr>
<td>Amazon EventBridge</td>
<td>Event rules</td>
<td>aws-ct-aft-capture-ct-events, aws-ct-aft-lambda-account-request-processor</td>
</tr>
<tr>
<td>Key Management Service (KMS)</td>
<td>Customer Managed Keys</td>
<td><em>.aws-ct-aft-</em></td>
</tr>
<tr>
<td>AWS Systems Manager</td>
<td>Parameter store</td>
<td>/aws-ct-aft/account/<em>, /aws/ct-aft/config/</em></td>
</tr>
<tr>
<td>Amazon SQS</td>
<td>Queues</td>
<td>aws-ct-aft-account-request.fifo, aws-ct-aft-account-request-dlg.fifo</td>
</tr>
<tr>
<td>CloudWatch</td>
<td>Log groups</td>
<td>/aws/<em>, /aws-ct-aft-</em></td>
</tr>
<tr>
<td>AWS Support Center (Optional)</td>
<td>Support plans</td>
<td>Enterprise</td>
</tr>
</tbody>
</table>

**AWS accounts provisioned through AWS Control Tower Account Factory for Terraform**

<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>
</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* for more information.

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Control Tower management account</td>
<td>AWS Identity and Access Management</td>
<td>Roles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AWS Systems Manager</td>
<td>Parameter store</td>
</tr>
<tr>
<td></td>
<td>AWS Organizations (Optional)</td>
<td>Service Control Policies</td>
</tr>
<tr>
<td></td>
<td>CloudTrail (Optional)</td>
<td>Trails</td>
</tr>
<tr>
<td></td>
<td>AWS Support Center (Optional)</td>
<td>Support plans</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Control Tower log archive account</td>
<td>AWS Identity and Access Management</td>
<td>Roles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Key Management Service (KMS)</td>
<td>Customer Managed Keys</td>
</tr>
<tr>
<td></td>
<td>Amazon S3</td>
<td>Buckets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AWS Support Center (Optional)</td>
<td>Support plans</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AWS service</th>
<th>Resource type</th>
<th>Resource name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Control Tower audit account</td>
<td>AWS Identity and Access Management</td>
<td>Roles</td>
</tr>
<tr>
<td></td>
<td>Amazon S3</td>
<td>Buckets</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AWS Support Center (Optional)</td>
<td>Support plans</td>
</tr>
</tbody>
</table>
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 roles, AFT creates two essential roles: the AWSAFTAdmin role and the AWSAFTExecution role, which 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"
        }
    ]
}
```

The following 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 uses the AWSControlTowerExecution role initially, to create the AWSAFTExecution role in specified accounts. The AWSAFTExecution role allows the AFT pipeline to run the steps performed during the AFT framework's provisioning and provisioning customizations stages, for AFT provisioned accounts.

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

**AdministratorAccess – AWS managed policy**

The following artifact shows the trust relationship for the AWSAFTExecution role. The placeholder number 012345678901 is replaced by the AFT management account ID number.
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.

- **Amazon Simple Notification Service** – AFT creates Amazon Simple Notification Service (SNS) topics in the AFT management account, which stores success and failure notifications after processing every AFT account request. You may receive these messages using your choice of protocol.

- **Amazon Simple Queuing Service** – AFT creates an Amazon Simple Queuing Service (Amazon SQS) FIFO queue in the AFT management account. The queue allows you to submit multiple account requests in parallel, but it sends one request at a time to AWS Control Tower Account Factory, for sequential processing.

- **AWS CodeBuild** – AFT creates AWS CodeBuild build projects in the AFT management account to initialize, compile, test, and apply Terraform plans for AFT source code in various build stages.

- **AWS CodePipeline** – AFT creates AWS CodePipeline pipelines in the AFT management account to integrate with your selected, supported AWS CodeStar connections provider for AFT source code, and to trigger build jobs in AWS CodeBuild.

- **AWS Lambda** – AFT creates AWS Lambda functions and layers in the AFT management account to perform steps during the account request, AFT account provisioning, and account customizations processes.

- **AWS Systems Manager Parameter Store** – AFT sets up the AWS Systems Manager Parameter Store in the AFT management account, to store the configuration parameters required for the AFT pipeline processes.

- **Amazon CloudWatch** – AFT creates Amazon CloudWatch log groups in the AFT management account to store logs generated by AWS services employed by the AFT pipeline. The retention period for CloudWatch logs is set to Never Expire.

- **Amazon VPC** – AFT creates an Amazon Virtual Private Cloud (VPC) to isolate services and resources in the AFT management account into a separate networking environment, for enhanced security.

- **AWS KMS** – AFT uses the AWS Key Management Service (KMS) in the AFT management account and in the AWS Control Tower log archive account. AFT creates keys to encrypt Terraform states, data stored
in DynamoDB tables, and SNS topics. These logs and artifacts are generated when AWS resources and services are deployed by AFT. KMS keys created by AFT have yearly rotation enabled by default.

- **AWS Identity and Access Management (IAM)** – AFT follows the recommended Least Privilege model. It creates AWS Identity and Access Management (IAM) roles and policies in the AFT management account, in AWS Control Tower accounts, and in AFT provisioned accounts, as needed, to perform actions required during the AFT pipeline workflow.

- **AWS Step Functions** – AFT creates AWS Step Functions state machines in the AFT management account. These state machines orchestrate and automate the process and steps for the AFT account provisioning framework and customizations.

- **Amazon EventBridge** – AFT creates an Amazon EventBridge event bus in the AFT and AWS Control Tower management account to capture and store AWS Control Tower lifecycle events long-term in the AFT management account’s DynamoDB table. AFT creates AWS CloudWatch event rules in the AFT management and AWS Control Tower management accounts, which trigger multiple steps required during running of the AFT pipeline workflow.

- **AWS CloudTrail (Optional)** – When this feature is enabled, AFT creates an AWS CloudTrail organization trail in the AWS Control Tower management account, for logging data events for Amazon S3 buckets and AWS Lambda functions. AFT sends these logs to a central S3 bucket in the AWS Control Tower log archive account.

- **AWS Support (Optional)** – When this feature is enabled, AFT turns on the AWS Enterprise Support plan for accounts provisioned by AFT. By default, AWS accounts are created with the AWS Basic Support plan enabled.

## 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. 141) 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 a pipeline in the AFT management account, which will run to perform the (next, global) Account customizations (p. 141) 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. 140). 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. 126) for more information about AFT deployment.
- Set up a git repository in your environment for AFT account provisioning customizations. See Post-deployment steps (p. 127) 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. 143).

**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 guardrail 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. 126).
- Pre-populated git repositories for global customizations and account customizations in your environment. See Step 3: Populate each repository in Post-deployment steps (p. 127) 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

To apply account customizations, you must push a specific folder structure to your chosen repository.

**Note**
Account customizations are applied automatically, after the global customizations stage in the AFT pipeline.

**Step 1: Create a folder for specific account customization**

First, locate the sample `ACCOUNT_TEMPLATE` folder (provided by AFT) in your chosen repository. You can copy that sample to a new folder with the name of your specific account customization, depending upon your organizational requirements. For example, `prod-customizations` or `finance-customizations`.

**Step 2: Add configurations for your specific account customizations folder**

- If your custom configurations are in the form of Python programs or scripts, place those under `[/account_customizations_name]/api_helpers/python` folder in your repository.
- If your custom configurations are in the form of Bash scripts, place those under `[/account_customizations_name]/api_helpers` folder in your repository.
- If your custom configurations are in the form of Terraform, place those under `[/account_customizations_name]/terraform` folder in your repository.
- Refer to the account customizations README file for more details on creating custom configurations.

**Step 3: Refer to the specific account_customizations_name parameter in the account request input file**

The AFT account request file provides an input parameter where you can specify the account customizations you wish to apply, after each account has been provisioned by means of the AFT pipeline. Enter the name of your specific account customizations folder as the value of the `account_customizations_name` parameter, in your account request input file, as shown in Post-deployment steps (p. 127).

To apply similar account customizations to multiple accounts in your environment, you may refer to the `account_customizations_name` input parameter in multiple account requests.

Multiple specific customizations
You can create multiple folders in your account customizations repository, which can set up a variety of specific account customizations. For each specific account customization implementation that you require, follow the steps given.

**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.

```
{
  "include": [
    { "type": "all" },
    { "type": "ous", "target_value": [ "ou1","ou2"] },
    { "type": "tags", "target_value": [ {"key1": "value1"}, {"key2": "value2"}] },
    { "type": "accounts", "target_value": [ "acc1_ID","acc2_ID"] }
  ],
  "exclude": [
    { "type": "ous", "target_value": [ "ou1","ou2"] },
    { "type": "tags", "target_value": [ {"key1": "value1"}, {"key2": "value2"}] },
    { "type": "accounts", "target_value": [ "acc1","acc2"] }
  ]
}
```
As shown in this example event schema, you can choose accounts to include or exclude from the re-invoke process. You can filter by account ID, organizational unit (OU), and account tags. If you don’t apply any filters and include the statement "type": "all", the customization for all AFT-provisioned accounts is re-invoked.

After the event parameters have been filled out, the AWS Step Function runs. It invokes the corresponding customizations.

Note
AFT can invoke a maximum of 5 customizations at a time. The step function 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.

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 (p. 124).
- AFT account provisioning customizations. Sample code available. For more information on AFT account provisioning customizations, see Create your AFT account provisioning customizations state machine (p. 140).
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. 234).
• Preserve Terraform state configuration generated at the time of AFT deployment. For details, see Deploy AWS Control Tower Account Factory for Terraform (AFT) (p. 126).
• 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. 138) 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. 138) 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 section walks you through the steps to remove accounts from AFT, so they are no longer deployed or updated by the pipeline. For example, you may be closing accounts for retiring applications, isolating accounts that are compromised, or moving accounts from one organization to another. Removing an account from AFT is not the same as removing an account from AWS Control Tower. After an account is provisioned by AFT, it is removed from AWS Control Tower in the same way as any other AWS account.

Important
Removing an account from AFT is irreversible. It can result in a loss of state.

Step 1: Remove account from the git repository that stores your account requests
In the git repository where you store account requests, delete the account request for the account you intend to remove from AFT.

Step 2: Delete the account customizations pipeline
In the AFT management account, delete the AWS CodePipeline for the account you intend to remove from AFT. This pipeline has the account ID as the prefix in its name.

Step 3: Delete the Terraform workspace (For Terraform Cloud and Terraform Enterprise customers only)
Delete the global customizations and account customizations workspaces for the account you intend to remove from AFT.

Step 4: Delete the Terraform state from the Amazon S3 backend
In the AFT management account, delete all relevant folders inside S3 buckets for the account you intend to remove from AFT. In the examples that follow, replace the placeholder number 012345678901 with the AFT management account ID number.
If you chose Terraform OSS, you will find 3 folders for each account (related to customizations pipeline state, global and account customizations state) in the aft-backend-012345678901-primary-region and aft-backend-012345678901-secondary-region S3 buckets.

If you chose Terraform Cloud or Terraform Enterprise, you will find a folder for each account (related to customizations pipeline state) in the aft-backend-012345678901-primary-region and aft-backend-012345678901-secondary-region S3 buckets.

**Step 5: Delete account metadata**

In the AFT management account, delete the account-related metadata for the account you intend to remove from AFT. This metadata is stored in the parameter aft-request-metadata in the DynamoDB table.

**Note**

AFT uses this metadata as input to invoke downstream stages in the AFT pipeline, such as the AFT account provisioning framework and customizations, and for global and account customizations.

**AFT release history**

This page summarizes releases of the AWS Control Tower Account Factory for Terraform (AFT) functionality. For more details see the Releases file for this GitHub repository.

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 2022</td>
<td><strong>AFT release 1.3.0</strong> refactors Python source code, and moves Lambda Handlers to <code>/src</code> instead of within Terraform modules. Subsequent minor releases include bugfixes.</td>
</tr>
<tr>
<td>February 2022</td>
<td><strong>AFT release 1.2.0</strong> enables AFT to manage existing AWS Control Tower accounts that were created outside of AFT. By specifying the account in the <code>aft-account-request</code> repository, AFT discovers the existing account.</td>
</tr>
<tr>
<td>January 2022</td>
<td><strong>AFT release 1.1.0</strong> includes SSM Parameters in vended accounts, to allow for customer fields. You now can define Custom fields in the <code>aft-account-request</code> repository. The fields are located in the vended account under the namespace: <code>/aft/account-request/customer-fields/</code>. If you remove a custom field from the account request, the field also is removed from the SSM parameter store in the account.</td>
</tr>
<tr>
<td>November 2021</td>
<td>Initial release</td>
</tr>
</tbody>
</table>
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 guardrails. 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.

Member account administrators can (and as a best practice, they should) subscribe to the SNS drift notifications for specific accounts. For example, the aws-controltowers-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. 179).

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

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. 151).

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 by selecting Re-register OU on the OU page, to repair drift in Account Factory provisioned accounts, or SCP drift.

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 guardrails 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 guardrail enabled on the OU, verifying that the guardrail'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</th>
<th>Burst</th>
<th>Sustained</th>
<th>OU or Landing Zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>listOrganizationalUnits</td>
<td>8</td>
<td>5</td>
<td>1 per landing zone</td>
</tr>
<tr>
<td>listPoliciesForTarget</td>
<td>8</td>
<td>5</td>
<td>1 per registered OU</td>
</tr>
<tr>
<td>describePolicy</td>
<td>2</td>
<td></td>
<td>1 per managed SCP</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`.

- **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 Account Factory accounts with AWS Service Catalog (p. 118).

In particular, if you've made certain changes to your accounts by means of AWS 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. 152)
- Removed Member Account (p. 153)
- Unplanned Update to Managed SCP (p. 154)
- SCP Attached to Member Account (p. 156)
- SCP Attached to Managed OU (p. 154)
- SCP Detached from Managed OU (p. 155)
- Deleted Foundational OU (p. 156)

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.

AWS Control Tower does not look for drift regarding other services that work with the management account, including CloudTrail, CloudWatch, AWS SSO, AWS CloudFormation, AWS Config, and so forth.
No drift detection is available in child accounts, because these accounts are protected by preventive mandatory guardrails.

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.

```
{
  "Message" : "AWS Control Tower has detected that your member account 'account-email@example.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 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. 167).
- Updating the provisioned product in Account Factory. For more information, see Update and move account factory accounts with AWS Service Catalog (p. 119).
- Updating your landing zone (slower option). For more information, see Update Your Landing Zone (p. 36).

**Note**

If you have several individual accounts to update, also see this method for making updates with a script: Provisioning and updating accounts using script automation (p. 37).

- 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. 36).

**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 the provisioned product for each moved account, because re-register OU will not perform the update. For more information, see Update and move account factory accounts with AWS Service Catalog (p. 119).

**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. 36).
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.

Added Member Account

Adding an account is not technically drift. However, AWS Control Tower alerts you when an AWS Control Tower account is added to your AWS Control Tower organization. For example, an account may be added to your AWS Control Tower organization as part of the drift remediation process if a shared account, such as the audit account or log archive account, has been removed and must be replaced. The following example shows an Amazon SNS notification you may receive when this type of event is detected.

```json
{
  "Message": "AWS Control Tower has detected that the account '"account-email@amazon.com (012345678909)' has been added to organization 'o-123EXAMPLE'. For more information, including steps to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/add-account'",
  "ManagementAccountId": "012345678912",
  "OrganizationId": "o-123EXAMPLE",
  "DriftType": "AccountAddedToOrganization",
  "RemediationStep": "Update Account Factory Provisioned Product",
  "AccountId": "012345678909"
}
```

Resolution

No resolution is required, because adding a member account to an OU or enrolling an Account Factory account does not cause drift. If a shared account has been removed and re-added, it is a special case, and you may need to update that shared account or the Security OU. For information about updating Account Factory accounts, see Update and move account factory accounts with AWS Service Catalog (p. 119).

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 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 Service Catalog (p. 119).

- 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. 157).

**Note**

In AWS 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 AWS 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 guardrail 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)', attached to the registered organizational unit 'Security (ou-0123-1EXAMPLE)', 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" : "ServiceControlPolicyUpdated",
    "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. 167).
- Updating your landing zone (slower option). For more information, see Update Your Landing Zone (p. 36).

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. 36).

### SCP Attached to Managed OU

This type of drift can occur when an SCP for a guardrail 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
```
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. 167).
- Updating your landing zone (slower option). For more information, see Update Your Landing Zone (p. 36).

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. 36).

SCP Detached from Managed OU

This type of drift can occur when an SCP for a guardrail 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.

```
{
    "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-ou'",
    "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. 167).
- Updating your landing zone (slower option). If the drift is affecting a mandatory guardrail, 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. 36).

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 guardrail, 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. 36).

SCP Attached to Member Account

This type of drift can occur when an SCP for a guardrail 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": "ServiceControlPolicyAttachedToAccount",
  "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. 36).

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. 167).

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. 36).

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 somewhere 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" : "OrganizationalUnitDeleted",
"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. 157).

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 of 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. 150).

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 OU list page, but they are not governed by AWS Control Tower guardrails.

Creating an account

Accounts created outside of AWS Control Tower are referred to as Unenrolled. Accounts that belong to an organization that's registered with AWS Control Tower are displayed in the Accounts list page. Accounts that do not belong to a registered organization 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 Account Factory 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 "Resolving Drift" in Detect and resolve drift in AWS Control Tower (p. 148).

AWS Control Tower displays the names of OUs in the AWS Control Tower dashboard and displays Additional OUs in Account Factory. 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 guardrails might not be enforced.

For general information about drift, see "Resolving Drift" in *Detect and resolve drift in AWS Control Tower* (p. 148).

**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 guardrail 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)**

1. Sign in to the AWS Control Tower console at [https://console.aws.amazon.com/controltower](https://console.aws.amazon.com/controltower). You must sign in as an AWS Identity and Access Management (IAM) user 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 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](https://docs.aws.amazon.com/organizations/latest/userguide/gs/invite-account.html).

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 an IAM 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.

**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.

If you see a deleted OU displayed in the AWS Control Tower console, repair your landing zone to remove outdated entries.

**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.

If you see a deleted account displayed in the AWS Control Tower console, repair your landing zone to remove the outdated entry.

**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**.
Enable AWS Control Tower on organizations and accounts

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.

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 Organizational units 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.

The Accounts page lists all accounts in your organization, regardless of OU or enrollment status in AWS Control Tower. You can view and enroll accounts individually within the OUs, if the accounts meet the prerequisites for enrollment.

Topics

• Register an existing organizational unit with AWS Control Tower (p. 167)
• Enroll an existing AWS account (p. 106)

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 guardrails 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 guardrails 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 guardrails **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 guardrails.

• New accounts created in OUs that are not registered with AWS Control Tower are not bound by guardrails, unless you specifically **Enroll** those accounts into AWS Control Tower. See [Enroll an existing AWS account](#) 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 guardrails, 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](#).

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?](#) for information about moving accounts from one landing zone to another.

### Considerations for AWS SSO and existing organizations

• If AWS Single Sign-On (AWS SSO) is already set up, the AWS Control Tower home Region must be the same as the AWS SSO Region.

• AWS Control Tower does not delete an existing configuration.

• If AWS SSO is already enabled, and if you are using SSO 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 Single Sign-On (AWS SSO) customers](#).

### 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](#).
Enable a Landing Zone in Existing AWS Organizations

This video (7:48), getting started with AWS Control Tower for AWS Organizations, describes how to set up and enable an AWS Control Tower landing zone in existing AWS Organizations. 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

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. 170).

- 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 guardrails.

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. 74).

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 SSO login for newly enrolled member accounts.
- Updates existing enrolled member accounts to reflect your landing zone changes.
- Updates guardrails 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 guardrails 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, 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 guardrail's compliance status is computed over all OUs on which the guardrail is enabled, including nested OUs.
• 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.

Nested OUs and drift

In certain situations, drift can prevent the registration of nested OUs.

Expectations for drift and nested OUs

• You can enable guardrails on OUs with drifted parents, but not on drifted OUs directly.
• You are allowed to enable detective guardrails under a drifted OU, as long as it’s not a top-level drifted OU.
• Mandatory guardrails are enabled on top-level OUs only. Mandatory guardrails are skipped when you register a nested OU.
• One mandatory guardrail protects AWS Config resources; therefore, that guardrail 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 guardrail 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 guardrails.

Nested OUs and guardrails

When you enable a guardrail on a registered OU, preventive and detective guardrails have different behaviors.
Preventive guardrails

- Preventive guardrails are enforced on nested OUs.
- Mandatory preventive guardrails are enforced on all accounts under the OU and its nested OUs.
- Preventive guardrails affect all accounts and OUs nested under the target OU, even if those accounts and OUs are not registered.

Detective guardrails

- Nested OUs do not inherit detective guardrails automatically; these must be enabled separately.
- Detective guardrails are deployed only to registered accounts in your landing zone’s operating Regions.

Enabled guardrail states and inheritance

You can view inherited guardrails for each OU, on the **OU details** page.

Inherited status

- The status **Inherited** indicates that the guardrail is enabled by inheritance only, and it has not been applied directly to the OU.
- The status **Enabled** means the guardrail is enforced on this OU, regardless of its state on other OUs.
- The status **Failed** means the guardrail is not enforced on this OU, regardless of its state on other OUs.

**Note**

The status **Inherited** indicates that the guardrail 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 guardrails** table represents one enabled guardrail 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 that all other OUs are “nested” under.

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 guardrails that apply to their OU.
Register an existing OU

In the AWS Control Tower console, on the **Organizational units** page, you can view all of your organizations, including OUs that are registered with AWS Control Tower and those that are not registered. 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.

**To register an existing OU**

2. In the left-pane navigation menu, choose **Organizational units**.
3. On the **Organizational units** page, select the radio button next to the OU you want to register.
4. At the upper right, select **Register OU**.

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.
Effects 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 effects 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 guardrails 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 guardrails. 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. 148).

**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 move the account, you can see a result “1 of 0” accounts in your OU details page. Furthermore, you may have created another unrolled account in that unregistered OU. If there’s an unregistered account, the console may read “1 of 1” for the OU. It will seem that the single (newly created) account is enrolled, but in fact it is not. You must enroll the new account.

Create a new OU

To create a new OU in AWS Control Tower

1. Navigate to the OU list page by selecting Organizational units on the left.
2. Select Add an OU 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.

**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. 164).

Update existing OUs and accounts

When you perform a landing zone update, you must update your enrolled accounts to apply new guardrails 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 multiple accounts

2. In the left-pane navigation menu, choose Organizational units.
3. On the Organizational units page, choose a Registered OU to view the details page.
4. Under Details in the upper right, select Re-Register OU.

Effects of re-registering 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 (SSO) login for those new enrolled accounts.
- Enrolled accounts are re-enrolled into AWS Control Tower.
- Drift on any preventive guardrails applied to the OU is fixed.
- All accounts are updated to reflect the latest landing zone changes.

For more information, see Enroll an existing AWS account (p. 106).

Note
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.

To update a single account

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 https://docs.aws.amazon.com/servicecatalog/latest/adminguide/productmgmt-update.html.

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. 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, by using the Enroll account form in the AWS Control Tower console.

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.

- 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 in the AWS CloudFormation User Guide.

- Exceeds account limit
  
  AWS Control Tower limits each OU to 298 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 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.

- SSO conflict
  
  The AWS Control Tower home Region is not the same as the AWS Single Sign-On (AWS SSO) Region. If AWS SSO is already set up, the AWS Control Tower home region must be the same as the AWS SSO 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 AWS 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. 110).

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.
Guardrails in AWS Control Tower

A guardrail is a high-level rule that provides ongoing governance for your overall AWS environment. It's expressed in plain language. Through guardrails, AWS Control Tower implements preventive or detective controls that help you govern your resources and monitor compliance across groups of AWS accounts.

A guardrail applies to an entire organizational unit (OU), and every AWS account within the OU is affected by the guardrail. Therefore, when users perform work in any AWS account in your landing zone, they're always subject to the guardrails that are governing their account's OU.

The purpose of guardrails

Guardrails enable you to express your policy intentions. For example, if you enable the detective guardrail Detect Whether Public Read Access to Amazon S3 Buckets is Allowed on an OU, you can determine whether a user would be permitted to have read access to any S3 buckets for any accounts under that OU.

Guardrail behavior and guidance

Guardrails are categorized according to their behavior and their guidance.

The behavior of each guardrail is either preventive or detective. Guardrail guidance refers to the recommended practice for how to apply each guardrail to your OUs. The guidance of a guardrail is independent of whether its behavior is preventive or detective.

Guardrail behavior

- **Preventive** – A preventive guardrail ensures that your accounts maintain compliance, because it disallows actions that lead to policy violations. The status of a preventive guardrail is either enforced or not enabled. Preventive guardrails are supported in all AWS Regions.
- **Detective** – A detective guardrail detects noncompliance of resources within your accounts, such as policy violations, and provides alerts through the dashboard. The status of a detective guardrail is either clear, in violation, or not enabled. Detective guardrails apply only in those AWS Regions supported by AWS Control Tower.

Implementation of guardrail behavior

- The preventive guardrails are implemented using Service Control Policies (SCPs), which are part of AWS Organizations.
- The detective guardrails are implemented using AWS Config rules.
- Certain mandatory guardrails 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 guardrail reference, under each mandatory guardrail to which that SCP applies.

Guardrail guidance

AWS Control Tower provides three categories of guidance: mandatory, strongly recommended, and elective guardrails.

- Mandatory guardrails are always enforced.
• Strongly recommended guardrails are designed to enforce some common best practices for well-architected, multi-account environments.
• Elective guardrails enable you to track or lock down actions that are commonly restricted in an AWS enterprise environment.

Defaults: When you create a new landing zone, AWS Control Tower enables all mandatory guardrails by default and applies them to your top-level OUs. When you extend governance to an OU, AWS Control Tower applies mandatory guardrails to the OU by default. Strongly recommended and elective guardrails are not enabled by default.

Considerations for guardrails and OUs

When working with guardrails and OUs, consider the following properties:

Guardrails, landing zones, and OUs

• After you create your landing zone, all resources in your landing zone, for example, Amazon S3 buckets, are subject to guardrails.
• OUs created through AWS Control Tower have mandatory guardrails applied to them automatically, and optional guardrails applied at the discretion of administrators.
• OUs created outside of an AWS Control Tower landing zone (that is, unregistered OUs) are displayed in the AWS Control Tower console, but AWS Control Tower guardrails do not apply to them, unless they become registered OUs.
• When you enable guardrails on an organizational unit (OU) that is registered with AWS Control Tower, preventive guardrails apply to all member accounts under the OU, enrolled and unenrolled. Detective guardrails apply to enrolled accounts only.

For more information about how guardrails are applied to nested OUs, in AWS Control Tower, see Nested OUs and guardrails (p. 166).

Exception to guardrails for the management account

The root user and any IAM administrators in the management account can perform work that guardrails 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 guardrails and accounts

When working with guardrails and accounts, consider the following properties:

Guardrails and accounts

• Accounts created through the Account Factory in AWS Control Tower inherit the guardrails 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 guardrails. These are called unenrolled accounts.
• Accounts created outside of AWS Control Tower won't inherit guardrails in AWS Control Tower until you enroll them. However, these unenrolled accounts are displayed in AWS Control Tower.

  Accounts inherit guardrails from an OU upon enrollment in that OU.
  • An OU can contain enrolled or unenrolled member accounts.
  • Guardrails do not apply to an unenrolled account unless it becomes a member account of a registered AWS Control Tower OU. In that case, preventive guardrails for the OU will apply to the unenrolled account. Detective guardrails will not apply.
  • When you enable optional guardrails, 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 guardrails entering an unknown state. For more information, see Guardrail reference (p. 181).

View guardrail details

To view details about an individual guardrail, select the name of the guardrail from the table on the Guardrails page.

In the guardrail details page of the console, you can find the following details for each guardrail:

• Name – The name of the guardrail.
• Description – A description of the guardrail.
• Guidance – The guidance is either mandatory, strongly recommended, or elective.
• Category – The guardrail category can be Audit logs, Data residency, Data security, Monitoring, Network, Operations, IAM, or AWS Control Tower Setup.
• Release date – The date the guardrail became available.
• Behavior – A guardrail’s behavior is set to either preventive or detective.
• Compliance Status – A guardrail’s compliance status can be Clear, Compliant, Enforced, Unknown, or In violation. For more information, see AWS Control Tower guardrail compliance status (p. 178).
• State – The state of enablement for a guardrail on the current OU. A guardrail’s state can be Enabled, Enabling, Failed to enable, Failed to disable, or Disabling. If a guardrail is enabled on an OU higher on the hierarchy tree of a nested OU, the state will show Inherited.
• The status of the Region deny guardrail is shown as a separate entry.

The guardrail 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 guardrail’s artifact, select Service control policy (SCP) or AWS Config rule on the Guardrail details page.

Enable guardrails

Most guardrails are enabled automatically according to an OU’s configuration, and some guardrails can be enabled manually on your OUs. The following procedure describes the steps for enabling guardrails on an OU.

Important

When you enable optional guardrails, 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 guardrails entering an unknown state.
To enable guardrails in an OU

2. From the left navigation, choose Guardrails.
3. Choose a guardrail that you want to enable; for example, Guardrail: Detect Whether Encryption is Enabled for Amazon EBS Volumes Attached to Amazon EC2 Instances. This choice opens the guardrail's details page.
4. From Organizational units enabled, choose Enable guardrail 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 guardrail.
6. Choose Enable guardrail on OU.
7. Your guardrail is now enabled. It may take several minutes for the change to complete. When it does, you'll see that this guardrail is applied to the OU you selected.

Note
You can enable one preventive guardrail at a time, but you can enable detective guardrails concurrently.

Concurrent detective guardrail deployment

When applying detective guardrails, you can deploy more than one guardrail at a time, up to the StackSets limitation of 5000 concurrent operations.

Functionality available

- Apply different detective guardrails on the same OU, concurrently. 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.
- Apply different detective guardrails on different OUs, concurrently.

Optional guardrails

Strongly recommended and elective guardrails 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 guardrails are not enabled by default. For more information about optional guardrails, see the following guardrail references:

- Strongly recommended guardrails (p. 195)
- Elective guardrails (p. 204)

Guardrails and compliance

Within AWS Control Tower, compliance means that cloud administrators know when the accounts in their organization are compliant with established policies, while builders can provision new AWS accounts quickly in a few clicks. AWS Control Tower guardrails embody the rules of compliance, so you can identify compliant and non-compliant resources. This page describes guardrail compliance status in detail.

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.

For more information about how AWS Control Tower helps you maintain compliance with governmental regulations and industry standards, see Compliance Validation.

Examples of compliance rules (guardrails) in AWS Control Tower:

- Detect Whether Public Write Access to Amazon S3 Buckets is Allowed (p. 200)
- Detect Whether Unrestricted Incoming TCP Traffic is Allowed (p. 197)

Examples of governmental compliance regulations:

- The U.S. Health Insurance Portability and Accountability Act of 1996 (HIPAA)
- The European Union's General Data Protection Regulation of 2016 (GDPR)

For ongoing governance, administrators can enable pre-configured guardrails—clearly defined rules for security, operations, and compliance. These guardrails can:

- prevent deployment of resources that don't conform to policies (by means of preventive guardrails, implemented with SCPs)
- continuously monitor deployed resources for nonconformance (by means of detective guardrails, implemented with AWS Config Rules)

How can administrators review compliance?

Compliance with detective guardrails is determined according to data retrieved from the AWS Config aggregator in the AWS Control Tower Audit account.

Detective guardrail status

To view the compliance status of detective guardrails, navigate to the Enrolled accounts table in the AWS Control Tower console. Accounts may show a compliance status of Unknown if any detective guardrails are misconfigured, which occurs most often due to Moved account drift.

Preventive guardrail status

The compliance status of preventive guardrails on an OU may be viewed on the OU page. If any preventive guardrails are misconfigured for an OU, the State field shows Registration failed status. Preventive guardrail misconfiguration is caused most often by SCP drift, which can occur if the guardrail's SCP is modified or detached from the OU by means of the AWS Organizations console.

The status also can be viewed in the organizational units table, on any page that has an organizational units table:

- the AWS Control Tower dashboard
- the OUs list
- the guardrail details page

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 Guardrail compliance notifications by SNS (p. 180), 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 guardrail compliance violation. For more information about drift, see Detect and resolve drift in AWS Control Tower (p. 148).

**AWS Control Tower guardrail compliance status**

This section lists the possible categories of compliance and non-compliance in AWS Control Tower.

**In violation** – Denotes that resources are actively breaching a compliance rule.

- **Applies to:** Detective guardrails (AWS Config Rules)
- **Reported for:** A guardrail across multiple accounts

**Enforced** – Maximum level of protection. Operations that would break this compliance rule are simply not allowed.

- **Applies to:** Preventive guardrails (SCPs)
- **Reported for:** A guardrail across multiple accounts

**Clear** – Compliance rules are properly in place. No violations have been detected.

- **Applies to:** Detective guardrails (AWS Config Rules)
- **Reported for:** A guardrail across multiple accounts

**Compliant** – Compliance rules are properly in place. No violations have been detected.

- **Applies to:** Detective guardrails (AWS Config Rules)
- **Reported for:**
  - A guardrail for a single account
  - An account across multiple guardrails
  - An OU across multiple accounts

**Noncompliant** – Compliance rules are properly in place. However, non-compliant resources have been detected.
**Drift prevention and notification**

You can enable certain guardrails 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 guardrails.

- **Passive protection**: AWS CloudTrail monitors and logs preventive guardrail (SCP) drift.
- **Active protection**: The AWS Control Tower [drift monitoring service](#) actively scans the preventive guardrail SCPs, on a regular basis.

AWS Control Tower notifies you by means of SNS messaging, if drift is detected.

**Drift prevention**

Some guardrails prevent modification of compliance reporting mechanisms.

- **Disallow Changes to AWS Config Rules Set Up by AWS Control Tower** *(Mandatory, preventive guardrail)*
- **Disallow Deletion of AWS Config Aggregation Authorizations Created by AWS Control Tower** *(Mandatory, preventive guardrail)*
- **Disallow Changes to Tags Created by AWS Control Tower for AWS Config Resources** *(Mandatory, preventive guardrail)*
- **Disallow Configuration Changes to AWS Config** *(Mandatory, preventive guardrail)*

In contrast to preventive guardrails, detective guardrails notify you of resources that violate the associated AWS Config rule.

**To receive SNS notifications about drift and guardrail compliance**
For information about how to receive appropriate drift and guardrail compliance notifications by Amazon SNS, see Guardrail compliance notifications by SNS (p. 180).

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.

Guardrail compliance notifications by SNS

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.
- 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 noncompliance 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.

---

**Guardrail reference**

The following sections include a reference for each of the guardrails available in AWS Control Tower. Each guardrail reference includes the details, artifacts, additional information, and considerations to keep in mind when enabling a specific guardrail on a OU in your landing zone.

**Topics**

- Mandatory guardrails (p. 181)
- Strongly recommended guardrails (p. 195)
- Elective guardrails (p. 204)

**Note**

The four mandatory guardrails with "Sid": "GRCLOUDTRAILENABLED" are identical by design. The sample code is correct.

**Mandatory guardrails**

Mandatory guardrails are enabled by default when you set up your landing zone and can't be disabled. Following, you'll find a reference for each of the mandatory guardrails available in AWS Control Tower.

**Topics**

- Disallow Changes to Encryption Configuration for AWS Control Tower Created Amazon S3 Buckets in Log Archive (p. 182)
- Disallow Changes to Logging Configuration for AWS Control Tower Created Amazon S3 Buckets in Log Archive (p. 183)
- Disallow Changes to Bucket Policy for AWS Control Tower Created Amazon S3 Buckets in Log Archive (p. 183)
• Disallow Changes to Lifecycle Configuration for AWS Control Tower Created Amazon S3 Buckets in Log Archive (p. 184)
• Disallow Changes to Amazon CloudWatch Logs Log Groups set up by AWS Control Tower (p. 184)
• Disallow Deletion of AWS Config Aggregation Authorizations Created by AWS Control Tower (p. 185)
• Disallow Deletion of Log Archive (p. 185)
• Detect Public Read Access Setting for Log Archive (p. 186)
• Detect Public Write Access Setting for Log Archive (p. 186)
• Disallow Configuration Changes to CloudTrail (p. 187)
• Integrate CloudTrail Events with Amazon CloudWatch Logs (p. 187)
• Enable CloudTrail in All Available Regions (p. 188)
• Enable Integrity Validation for CloudTrail Log File (p. 188)
• Disallow Changes to Amazon CloudWatch Set up by AWS Control Tower (p. 189)
• Disallow Changes to Tags Created by AWS Control Tower for AWS Config Resources (p. 189)
• Disallow Configuration Changes to AWS Config (p. 190)
• Enable AWS Config in All Available Regions (p. 190)
• Disallow Changes to AWS Config Rules Set Up by AWS Control Tower (p. 191)
• Disallow Changes to AWS IAM Roles Set Up by AWS Control Tower and AWS CloudFormation (p. 191)
• Disallow Changes to AWS Lambda Functions Set Up by AWS Control Tower (p. 193)
• Disallow Changes to Amazon SNS Set Up by AWS Control Tower (p. 194)
• Disallow Changes to Amazon SNS Subscriptions Set Up by AWS Control Tower (p. 194)

Note
The four mandatory guardrails with "Sid": "GRCTAUDITBUCKETENCRYPTIONCHANGESPROHIBITED" 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 guardrail prevents changes to encryption for the Amazon S3 buckets that AWS Control Tower creates in the log archive account. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on the Security OU. It cannot be enabled on additional OUs.

The artifact for this guardrail is the following service control policy (SCP).

```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 guardrail prevents changes to logging configuration for the Amazon S3 buckets that AWS Control Tower creates in the log archive account. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on the Security OU. It cannot be enabled on additional OUs.

The artifact for this guardrail is the following SCP.

```
{
    "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 guardrail prevents changes to bucket policy for the Amazon S3 buckets that AWS Control Tower creates in the log archive account. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on the Security OU. It cannot be enabled on additional OUs.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCTAUDITBUCKETPOLICYCHANGESPROHIBITED",
            "Effect": "Deny",
            "Action": [ "s3:PutBucketPolicy", "s3:DeleteBucketPolicy" ],
            "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 guardrail prevents lifecycle configuration changes for the Amazon S3 buckets that AWS Control Tower creates in the log archive account. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on the Security OU. It cannot be enabled on additional OUs.

The artifact for this guardrail 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 guardrail 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 guardrail with mandatory guidance. By default, this guardrail is enabled on all OUs.

The artifact for this guardrail is the following SCP.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRLOGGROUPPOLICY",
         "Effect": "Deny",
         "Action": [ "logs:DeleteLogGroup", "logs:PutRetentionPolicy" ],
         "Resource": [ "arn:aws:logs:*::*:log-group:*aws-controltower*" ],
         "Condition": {
            "StringNotLike": {
               "aws:PrincipalArn": [ "arn:aws:iam::*:role/AWSControlTowerExecution"
            }
         }
      }
   ]
}
```
Disallow Deletion of AWS Config Aggregation Authorizations Created by AWS Control Tower

This guardrail 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 guardrail with mandatory guidance. By default, this guardrail is enabled on all OUs.

The artifact for this guardrail 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 guardrail prevents deletion of Amazon S3 buckets created by AWS Control Tower in the log archive account. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on the Security OU.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRAUDITBUCKETDELETIONPROHIBITED",
            "Effect": "Deny",
            "Action": [
                "s3:DeleteBucket"
            ],
            "Resource": [
                "arn:aws:s3:::aws-controltower*"
            ],
            "Condition": {
```

185
Detect Public Read Access Setting for Log Archive

This guardrail detects whether public read access is enabled to the Amazon Amazon S3 buckets in the log archive shared account. This guardrail does not change the status of the account. This is a detective guardrail with mandatory guidance. By default, this guardrail is enabled on the Security OU.

The artifact for this guardrail 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 Public Write Access Setting for Log Archive

This guardrail detects whether public write access is enabled to the Amazon Amazon S3 buckets in the log archive shared account. This guardrail does not change the status of the account. This is a detective guardrail with mandatory guidance. By default, this guardrail is enabled on the Security OU.

The artifact for this guardrail 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
```
Mandatory guardrails

SourceIdentifier: S3_BUCKET_PUBLIC_WRITE_PROHIBITED
Scope:
  ComplianceResourceTypes:
  - AWS::S3::Bucket

Disallow Configuration Changes to CloudTrail

This guardrail prevents configuration changes to CloudTrail in your landing zone. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "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 guardrail performs real-time analysis of activity data by sending CloudTrail events to CloudWatch Logs log files. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "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 CloudTrail in All Available Regions

This guardrail enables CloudTrail in all available AWS Regions. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail 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 guardrail 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 guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail 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 guardrail disallows changes to Amazon CloudWatch as it was configured by AWS Control Tower when you set up your landing zone. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRCLOUDWATCHEVENTPOLICY",
         "Effect": "Deny",
         "Action": [
            "events:PutRule",
            "events:PutTargets",
            "events:DisableRule",
            "events:DeleteRule"
         ],
         "Resource": ["arn:aws:events:*:*:rule/aws-controltower-*"]
      }
   ]
}
```

Disallow Changes to Tags Created by AWS Control Tower for AWS Config Resources

This guardrail 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 guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "GRCONFIGRULETAGSPOLICY",
         "Effect": "Deny",
         "Action": [
            "config:TagResource",
            "config:UntagResource"
         ],
         "Resource": ["*"]
      }
   ]
}
```
Disallow Configuration Changes to AWS Config

This guardrail 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 guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail 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": ["*"]
    }
  ]
}
```

Enable AWS Config in All Available Regions

This guardrail enables AWS Config in all available AWS Regions. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRCONFIGENABLED",
      "Effect": "Deny",
      "Action": [
        "config:DeleteConfigurationRecorder",
        "config:DeleteDeliveryChannel",
        "config:DeleteRetentionConfiguration",
        "config:PutConfigurationRecorder",
        "config:PutDeliveryChannel"
      ],
      "Resource": ["*"]
    }
  ]
}
```
Disallow Changes to AWS Config Rules Set Up by AWS Control Tower

This guardrail disallows changes to AWS Config Rules that were implemented by AWS Control Tower when the landing zone was set up. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRCONFIGRULEPOLICY",
      "Effect": "Deny",
      "Action": [
        "config:PutConfigRule",
        "config:DeleteConfigRule",
        "config:DeleteEvaluationResults",
        "config:DeleteConfigurationAggregator",
        "config:GetConfigurationAggregator"
      ],
      "Resource": ["*"],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
  ]
}
```

Disallow Changes to AWS IAM Roles Set Up by AWS Control Tower and AWS CloudFormation

This guardrail disallows changes to the AWS IAM roles that AWS Control Tower created when the landing zone was set up. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

Guardrail update

An updated version has been released for the mandatory guardrail AWS-GR_IAM_ROLE_CHANGE_PROHIBITED.
This change to the guardrail 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 guardrail prevents this role from being created.

AWS Control Tower updated the existing guardrail to add an exception so that AWS CloudFormation StackSets can create the AWSControlTowerExecution role. As a second measure, this new guardrail protects the StackSets role to prevent principals in the child account from gaining access.

The new guardrail 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 guardrail version affects your OUs and accounts as follows:

- If you extend governance to an OU, that incoming OU receives the updated version of the guardrail 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 guardrail 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 guardrail, which blocks the creation of the AWSControlTowerExecution role.
- One consequence of this guardrail update is that your OUs can be functioning with different versions of the guardrail. Update your landing zone to apply the updated version of the guardrail to your OUs uniformly.

The artifact of the updated guardrail 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:GetRolePermissionsBoundary",
        "iam:GetRolePolicy",
        "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
    ]
  }
}
]}
]
]

The former artifact for this guardrail is the following SCP.

```
{
  "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*"
      ],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
  ]
}
```

Disallow Changes to AWS Lambda Functions Set Up by AWS Control Tower

This guardrail disallows changes to AWS Lambda functions set up by AWS Control Tower. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRLAMBDAFUNCTIONPOLICY",
      "Effect": "Deny",
      "Action": [ 
        "lambda:AddPermission",
        "lambda:InvokeLambda",
        "lambda:ListProvisionedConcurrencyExecutions",
        "lambda:ListProvisionedConcurrencySettings",
        "lambda:PutProvisionedConcurrencyExecution",
        "lambda:PutPermission",
        "lambda:UpdateFunctionConfiguration",
        "lambda:UpdateFunctionSecurityGroupIngress"
      ],
      "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 guardrail 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 guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRSNSTOPICPOLICY",
      "Effect": "Deny",
      "Action": [
        "sns:AddPermission",
        "sns:CreateTopic",
        "sns:DeleteTopic",
        "sns:RemovePermission",
        "sns:SetTopicAttributes"
      ],
      "Resource": [
        "arn:aws:sns:*::*:aws-controltower-*"
      ],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
  ]
}
```

**Disallow Changes to Amazon SNS Subscriptions Set Up by AWS Control Tower**

This guardrail 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 guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail 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"
        }
      }
    }
  ]
}
```

Strongly recommended guardrails

Strongly recommended guardrails are based on best practices for well-architected multi-account environments. These guardrails are not enabled by default, and can be disabled. Following, you'll find a reference for each of the strongly recommended guardrails available in AWS Control Tower.

Topics

- Disallow Creation of Access Keys for the Root User (p. 195)
- Disallow Actions as a Root User (p. 196)
- Detect Whether Encryption is Enabled for Amazon EBS Volumes Attached to Amazon EC2 Instances (p. 196)
- Detect Whether Unrestricted Incoming TCP Traffic is Allowed (p. 197)
- Detect Whether Unrestricted Internet Connection Through SSH is Allowed (p. 199)
- Detect Whether MFA for the Root User is Enabled (p. 199)
- Detect Whether Public Read Access to Amazon S3 Buckets is Allowed (p. 200)
- Detect Whether Public Write Access to Amazon S3 Buckets is Allowed (p. 200)
- Detect Whether Amazon EBS Volumes are Attached to Amazon EC2 Instances (p. 201)
- Detect Whether Amazon EBS Optimization is Enabled for Amazon EC2 Instances (p. 202)
- Detect Whether Public Access to Amazon RDS Database Instances is Enabled (p. 203)
- Detect Whether Public Access to Amazon RDS Database Snapshots is Enabled (p. 203)
- Detect Whether Storage Encryption is Enabled for Amazon RDS Database Instances (p. 203)

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 with limited permissions to interact with your AWS account. This is a preventive guardrail with strongly recommended guidance. By default, this guardrail is not enabled.
The artifact for this guardrail is the following SCP.

```
{
    "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 AWS Identity and Access Management (IAM) users for everyday interaction with your AWS account. This is a preventive guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following SCP.

```
{
    "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 guardrail detects whether the Amazon EBS volumes attached to an Amazon EC2 instance are encrypted. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail isn't enabled on any OUs.
The artifact for this guardrail is the following AWS Config rule.

```
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 guardrail 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 guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail 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: 197
```
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 guardrail detects whether internet connections are allowed through remote services such as the Secure Shell (SSH) protocol. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```yaml
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.
      Scope:
        ComplianceResourceTypes:
          - AWS::EC2::SecurityGroup
      Source:
        Owner: AWS
        SourceIdentifier: INCOMING_SSH_DISABLED
```

Detect Whether MFA for the Root User is Enabled

This guardrail 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 guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail 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
```
Detect Whether Public Read Access to Amazon S3 Buckets is Allowed

This guardrail detects whether public read access is allowed to Amazon S3 buckets. It helps you maintain secure access to data stored in the buckets. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail 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 guardrail detects whether public write access is allowed to Amazon S3 buckets. It helps you maintain secure access to data stored in the buckets. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
```

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
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 guardrail detects whether an Amazon EBS volume device persists independently from an Amazon EC2 instance. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

AWS::TemplateFormatVersion: 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
    Description: 'Check for Delete on termination'
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:
Strongly recommended guardrails

ComplianceResourceTypes:
- AWS::EC2::Volume

Detect Whether Amazon EBS Optimization is Enabled 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 guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail 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 guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail 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.
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 guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail 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 guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail 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
**Elective guardrails**

Elective guardrails enable you to lock down or track attempts at performing commonly restricted actions in an AWS enterprise environment. These guardrails are not enabled by default, and can be disabled. Following, you'll find a reference for the elective guardrails available in AWS Control Tower. The elective guardrails specifically for data residency are collected into a separate section, Guardrails that enhance data residency protection (p. 209).

**Topics**

- Disallow Changes to Encryption Configuration for Amazon S3 Buckets [Previously: Enable Encryption at Rest for Log Archive] (p. 204)
- Disallow Changes to Logging Configuration for Amazon S3 Buckets [Previously: Enable Access Logging for Log Archive] (p. 205)
- Disallow Changes to Bucket Policy for Amazon S3 Buckets [Previously: Disallow Policy Changes to Log Archive] (p. 205)
- Disallow Changes to Lifecycle Configuration for Amazon S3 Buckets [Previously: Set a Retention Policy for Log Archive] (p. 206)
- Disallow Changes to Replication Configuration for Amazon S3 Buckets (p. 206)
- Disallow Delete Actions on Amazon S3 Buckets Without MFA (p. 207)
- Detect Whether MFA is Enabled for AWS IAM Users (p. 207)
- Detect Whether MFA is Enabled for AWS IAM Users of the AWS Console (p. 208)
- Detect Whether Versioning for Amazon S3 Buckets is Enabled (p. 209)
- Guardrails that enhance data residency protection (p. 209)

**Disallow Changes to Encryption Configuration for Amazon S3 Buckets [Previously: Enable Encryption at Rest for Log Archive]**

This guardrail disallows changes to encryption for all Amazon S3 buckets. This is a preventive guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following service control policy (SCP).

```json
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "GRAUDITBUCKETENCRYPTIONENABLED",
        "Effect": "Deny",
        "Action": ["s3:PutEncryptionConfiguration"],
        "Resource": ["*"],
        "Condition": {
```
Disallow Changes to Logging Configuration for Amazon S3 Buckets [Previously: Enable Access Logging for Log Archive]

This guardrail disallows changes to logging configuration for all Amazon S3 buckets. This is a preventive guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail 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 guardrail disallows changes to bucket policy for all Amazon S3 buckets. This is a preventive guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail 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 guardrail disallows lifecycle configuration changes for all Amazon S3 buckets. This is a preventive guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following SCP.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRAUDITBUCKETRETENTIONPOLICY",
      "Effect": "Deny",
      "Action": [
        "s3:PutLifecycleConfiguration"
      ],
      "Resource": ["*"]
    },
    { "ArnNotLike": {
      "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
    }
    }
  ]
}
```

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 guardrail prevents that replication setting from being changed. This is a preventive guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail 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 guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRRESTRICTS3DELETEWITHOUTMFA",
      "Effect": "Deny",
      "Action": [
        "s3:DeleteObject",
        "s3:DeleteBucket"
      ],
      "Resource": ["*"]
    },
    "Condition": {
      "BoolIfExists": {
        "aws:MultiFactorAuthPresent": ["false"]
      }
    }
  ]
}
```

Detect Whether MFA is Enabled for AWS IAM Users

This guardrail detects whether MFA is enabled for AWS IAM users. You can protect your account by requiring MFA for all AWS IAM users in the account. MFA requires an additional authentication code after the user name and password are successful. This guardrail does not change the status of the account. This is a detective guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
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
```
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 guardrail detects whether MFA is enabled. This guardrail does not change the status of the account. This is a detective guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```json
AWS::Config::Rule
  Name: CheckForIAMUserMFA
  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: AWS
  SourceIdentifier: IAM_USER_MFA_ENABLED
  MaximumExecutionFrequency:
    !FindInMap
    - Settings
      - FrequencyMap
      - !Ref MaximumExecutionFrequency
```

```json
AWS::Config::Rule
  Name: CheckForIAMUserConsoleMFA
  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: AWS
  SourceIdentifier: IAM_USER_MFA_ENABLED
  MaximumExecutionFrequency:
    !FindInMap
    - Settings
      - FrequencyMap
      - !Ref MaximumExecutionFrequency
```

```json
AWS::Config::Rule
  Name: CheckForIAMUserConsoleMFA
  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: AWS
  SourceIdentifier: IAM_USER_MFA_ENABLED
  MaximumExecutionFrequency:
    !FindInMap
    - Settings
      - FrequencyMap
      - !Ref MaximumExecutionFrequency
```
Elective guardrails

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 guardrail does not change the status of the account. This is a detective guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

Guardrails that enhance data residency protection

These elective guardrails complement your enterprise’s data residency posture. By applying these guardrails 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 guardrails 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 guardrails. 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 guardrails

This video (5:58) describes how to enable data residency controls with AWS Control Tower guardrails. 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.

Topics

- Deny access to AWS based on the requested AWS Region (p. 210)
- Disallow internet access for an Amazon VPC instance managed by a customer (p. 211)
- Disallow Amazon Virtual Private Network (VPN) connections (p. 212)
- Disallow cross-region networking for Amazon EC2, Amazon CloudFront, and AWS Global Accelerator (p. 213)
- Detect whether public IP addresses for Amazon EC2 autoscaling are enabled through launch configurations (p. 213)
- Detect whether replication instances for AWS Database Migration Service are public (p. 214)
- Detect whether Amazon EBS snapshots are restorable by all AWS accounts (p. 215)
- Detect whether any Amazon EC2 instance has an associated public IPv4 address (p. 216)
- Detect whether Amazon S3 settings to block public access are set as true for the account (p. 217)
- Detects whether an Amazon EKS endpoint is blocked from public access (p. 218)
- Detect whether an Amazon OpenSearch Service domain is in Amazon VPC (p. 219)
- Detect whether any Amazon EMR cluster master nodes have public IP addresses (p. 220)
- Detect whether the AWS Lambda function policy attached to the Lambda resource blocks public access (p. 221)
- Detect whether public routes exist in the route table for an Internet Gateway (IGW) (p. 221)
- Detect whether Amazon Redshift clusters are blocked from public access (p. 222)
- Detect whether an Amazon SageMaker notebook instance allows direct internet access (p. 222)
- Detect whether any Amazon VPC subnets are assigned a public IP address (p. 223)
- Detect whether AWS Systems Manager documents owned by the account are public (p. 224)

Deny access to AWS based on the requested AWS Region

This guardrail is commonly referred to as the Region deny guardrail.

This guardrail 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 guardrails. Those services are specified in the SCP example code that follows.

This is an elective guardrail with preventive guidance. It is the primary guardrail associated with the Region deny action. For more information, see Configure the Region deny guardrail (p. 101).

The format for this guardrail is based on the following SCP.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRREGIONDENY",
            "Effect": "Deny",
            "NotAction": [
                "*:"]
        }
    ]
}
```
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 guardrail 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 guardrail. 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. 272).

- This guardrail 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 guardrail is applied, access cannot be changed.

This is a preventive guardrail with elective guidance. By default, this guardrail isn't enabled on any OUs.

The artifact for this guardrail is the following service control policy (SCP).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRDISALLOWVPCINTERNETACCESS",
            "Effect": "Deny",
            "Action": [
                "ec2:CreateInternetGateway",
                "ec2:AttachInternetGateway",
                "ec2:CreateEgressOnlyInternetGateway",
                "ec2:AttachEgressOnlyInternetGateway",
                "ec2:CreateDefaultVpc",
                "ec2:CreateDefaultSubnet",
                "ec2:CreateCarrierGateway"
            ],
            "Resource": ["*"]
        },
        "Condition": {
            "ArnNotLike": {
                "aws:PrincipalArn": ["arn:aws:iam::*:role/AWSControlTowerExecution"
            ]
        }
    }
}
```

Disallow Amazon Virtual Private Network (VPN) connections
This guardrail 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 guardrail with elective guidance. By default, this guardrail isn't enabled on any OUs.

The artifact for this guardrail is the following service control policy (SCP).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRDISALLOWVPNCONNECTIONS",
            "Effect": "Deny",
```
Disallow cross-region networking for Amazon EC2, Amazon CloudFront, and AWS Global Accelerator

This guardrail 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 guardrail prevents Amazon EC2 VPC peering and Amazon EC2 transit gateway peering within a single Region, as well as across Regions. For this reason, this guardrail may affect certain workloads in addition to your data residency posture.

This is a preventive guardrail with elective guidance. By default, this guardrail isn't enabled on any OUs.

The artifact for this guardrail is the following service control policy (SCP).

```json
{
    "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 guardrail detects whether Amazon EC2 Auto Scaling groups have public IP addresses enabled through launch configurations.
This is a detective guardrail with elective guidance. By default, this guardrail 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 guardrail 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 guardrail detects whether AWS Database Migration Service replication instances are public.

This is a detective guardrail with elective guidance. By default, this guardrail 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 guardrail 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.
```
Elective guardrails

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:
      !FindInMap
      - Settings
      - FrequencyMap
      - !Ref MaximumExecutionFrequency

Detect whether Amazon EBS snapshots are restorable by all AWS accounts

This guardrail detects whether all AWS accounts have access to restore Amazon EBS snapshots.

This is a detective guardrail with elective guidance. By default, this guardrail 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 guardrail is the following AWS Config rule.

AWSControlTowerUserGuide
Elective guardrails

- 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:
    !FindInMap
    - Settings
    - FrequencyMap
    - !Ref MaximumExecutionFrequency

Detect whether any Amazon EC2 instance has an associated public IPv4 address

This guardrail detects whether an Amazon Elastic Compute Cloud (Amazon EC2) instance has an associated public IPv4 address. This guardrail applies only to IPv4 addresses.

This is a detective guardrail with elective guidance. By default, this guardrail 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 guardrail is the following AWS Config rule.

AWS::TemplateFormatVersion: 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:
Elective guardrails

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 guardrail 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 guardrail with elective guidance. By default, this guardrail isn't enabled on any OUs.

The artifact for this guardrail is the following AWS Config rule.

```json
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:
```

217
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 guardrail detects whether an Amazon Elastic Kubernetes Service (Amazon EKS) endpoint is blocked from public access.

This is a detective guardrail with elective guidance. By default, this guardrail 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 guardrail 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.
```
Elective guardrails

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 guardrail detects whether an Amazon OpenSearch Service domain is in Amazon VPC.

This is a detective guardrail with elective guidance. By default, this guardrail 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 guardrail is the following AWS Config rule.

```json
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
```
Detect whether any Amazon EMR cluster master nodes have public IP addresses

This guardrail detects whether any Amazon EMR cluster master nodes have public IP addresses.

This is a detective guardrail with elective guidance. By default, this guardrail 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 guardrail checks clusters that are in RUNNING or WAITING state.

The artifact for this guardrail is the following AWS Config rule.

```yaml
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 guardrail 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 guardrail detects whether the AWS Lambda function policy attached to the Lambda resource blocks public access.

This is a detective guardrail with elective guidance. By default, this guardrail 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 guardrail is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 2010-09-09
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.
      Scope:
        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 guardrail detects whether public routes exist in the route table associated with an Internet Gateway (IGW).

This is a detective guardrail with elective guidance. By default, this guardrail 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.

The artifact for this guardrail 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:
```
Detect whether Amazon Redshift clusters are blocked from public access

This guardrail detects whether Amazon Redshift clusters are blocked from public access.

This is a detective guardrail with elective guidance. By default, this guardrail 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 guardrail is the following AWS Config rule.

```
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 guardrail detects whether an Amazon SageMaker notebook instance allows direct internet access.
This is a detective guardrail with elective guidance. By default, this guardrail 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 guardrail is the following AWS Config rule.

```json
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rule to detect whether an Amazon SageMaker notebook instance allows direct internet 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:
  SagemakerNotebookNoDirectInternetAccess:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Detects whether direct internet access is allowed for an Amazon SageMaker notebook instance. The rule is NON_COMPLIANT if Amazon SageMaker notebook instances allow direct internet access.
      Source:
        Owner: AWS
        SourceIdentifier: SAGEMAKER_NOTEBOOK_NO_DIRECT_INTERNET_ACCESS
      MaximumExecutionFrequency:
        !FindInMap
        - Settings
        - FrequencyMap
        - !Ref MaximumExecutionFrequency
```

**Detect whether any Amazon VPC subnets are assigned a public IP address**

This guardrail detects whether Amazon Virtual Private Cloud (Amazon VPC) subnets are assigned a public IP address.

This is a detective guardrail with elective guidance. By default, this guardrail 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 guardrail is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 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 guardrail detects whether AWS Systems Manager documents owned by the account are public.

This is a detective guardrail with elective guidance. By default, this guardrail 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 guardrail is the following AWS Config rule.

```yaml
AWSTemplateFormatVersion: 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
  12hours : Twelve_Hours
  24hours : TwentyFour_Hours

Resources:
SsmDocumentNotPublic:
  Type: AWS::Config::ConfigRule
  Properties:
    ConfigRuleName: !Sub ${ConfigRuleName}
    Description: Detects whether AWS Systems Manager (SSM) documents owned by the account are public. This rule is NON_COMPLIANT if any documents with owner 'Self' are public.
    Source:
      Owner: AWS
      SourceIdentifier: SSM_DOCUMENT_NOT_PUBLIC
  MaximumExecutionFrequency:
    !FindInMap
    - Settings
    - FrequencyMap
    - !Ref MaximumExecutionFrequency
Integrated services

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.

Topics
- Scripting Environments with AWS CloudFormation (p. 226)
- Monitoring Events with CloudTrail (p. 226)
- Monitoring Resources and Services with CloudWatch (p. 227)
- Govern Resource Configurations with AWS Config (p. 227)
- Manage Permissions for Entities with IAM (p. 227)
- Run Serverless Compute Functions with Lambda (p. 227)
- Manage Accounts Through AWS Organizations (p. 227)
- Store Objects with Amazon S3 (p. 228)
- Provisioning Accounts Through AWS Service Catalog (p. 228)
- Managing Users and Access Through AWS Single Sign-On (p. 229)
- Tracking Alerts Through Amazon Simple Notification Service (p. 232)
- Build Distributed Applications with AWS Step Functions (p. 233)

Scripting 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 guardrails on accounts.

Monitoring Events with CloudTrail

With AWS CloudTrail, you can monitor your AWS environment in the cloud by getting 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 the calls were made from, and 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.

AWS Control Tower sets up a new trail when you set up a landing zone. AWS Control Tower configures CloudTrail to enable centralized logging and auditing. It can be used in the management account to review administrative actions and lifecycle events.

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.

Monitoring 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.

AWS Control Tower uses AWS Config Rules with detective guardrails. For more information, see Guardrails in AWS Control Tower (p. 173).

Manage Permissions for Entities with IAM

AWS Identity and Access Management (IAM) is a web service for securely controlling access to AWS services. With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users and applications can access.

When you set up your landing zone, a number of groups are created for AWS SSO. These groups have permission sets that are pre-defined permissions policies from IAM. Your end users can also use IAM to define the scope of permissions for IAM users and other entities within member accounts.

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 guardrails 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 guardrails to it and its accounts. For information on registering an OU, see *Register an existing organizational unit with AWS Control Tower (p. 167)*.

**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.

** Provisioning Accounts Through AWS Service Catalog**

AWS Service Catalog enables IT administrators to create, manage, and distribute portfolios of approved products to end users, who then have access 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 *AWS 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 AWS Service Catalog. For more information, see *Provision and manage accounts with Account Factory (p. 117)*.
AWS Control Tower also can make use of the AWS Service Catalog APIs to further automate account provisioning and updating. For details, see the AWS Service Catalog Developer Guide.

Managing Users and Access Through AWS Single Sign-On

AWS Single Sign-On is a cloud-based service that simplifies how you manage SSO access to AWS accounts and business applications. You can control SSO access and user permissions across all your AWS accounts in AWS Organizations. You also can administer access to popular business applications and custom applications that support Security Assertion Markup Language (SAML) 2.0. Also, AWS SSO 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 Single Sign-On User Guide.

Working With AWS SSO and AWS Control Tower

In AWS Control Tower, AWS Single Sign-On allows central cloud administrators and end users to manage access to multiple AWS accounts and business applications. AWS Control Tower uses this service to set up and manage access to the accounts created through AWS Service Catalog.

For a brief tutorial about how to set up your SSO 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 SSO in AWS Control Tower.

About setting up AWS Control Tower with AWS SSO

When you initially set up AWS Control Tower, only the root user and any IAM users with the correct permissions can add AWS SSO users. However, after end users have been added in the AWSAccountFactory group, they can create new SSO users from the Account Factory wizard. For more information, see Provision and manage accounts with Account Factory (p. 117).

Your landing zone is set up 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.

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 AWS 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. 29)

For more information on how to use this service in the context of AWS Control Tower, see the following topics in the AWS Single Sign-On 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 AWS SSO directory in your home region. If you set up your landing zone in another Region and then navigate to the AWS SSO console, you must change the Region to your home region. Do not delete your AWS SSO configuration in your home region.

**Things to Know About SSO Accounts and AWS Control Tower**

Here are some good things to know when working with AWS SSO user accounts in AWS Control Tower.

• If your AWS SSO user account is disabled, you'll get an error message when trying to provision new accounts in Account Factory. You can re-enable your SSO user in the AWS SSO console.
• If you specify a new SSO 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 SSO user account. The previously created user account is not removed. If you prefer to remove the previous SSO user email address from AWS SSO, see Disabling a User.
• AWS SSO 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 SSO and different identity sources, refer to the Considerations for Changing Your Identity Source in the AWS SSO documentation.

**AWS SSO 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 AWS SSO. Doing so matches permission sets to users in groups within your accounts. The groups created when you set up your landing zone are as follows.

**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>
</tbody>
</table>
### AWS Control Tower User Guide

#### Amazon SNS

<table>
<thead>
<tr>
<th>Account</th>
<th>Permission sets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>

### Tracking 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 [Amazon Simple Notification Service Developer Guide](https://docs.aws.amazon.com/sns/latest/dg/).

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. 148).

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 guardrail 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 *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 user accounts with 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 recommend TLS 1.2 or later.
- 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 personal 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 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 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. 235)
- Encryption in Transit (p. 235)
- Restrict Access to Content (p. 235)

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 AWS SSO 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 AWS SSO groups that are created when you set up your landing zone, see AWS SSO Groups for AWS Control Tower (p. 230).

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 Single Sign-On (AWS SSO) require that 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 user name and a password.

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. 236)
- Access Control (p. 237)
- Overview of Managing Access Permissions to Your AWS Control Tower Resources (p. 237)
- Prevent cross-service impersonation (p. 240)
- Using Identity-Based Policies (IAM Policies) for AWS Control Tower (p. 240)

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 user. 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 Sign in as a Root User (p. 30).

- **IAM user** – An IAM user is an identity within your AWS account that has specific custom permissions. You can use an IAM user name and password to sign in to secure AWS webpages like the AWS Management Console, AWS Discussion Forums, or the AWS Support Center.

  In addition to a user name and password, you can also generate access keys for each user. You can use these keys when you access 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 with 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 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.

- **AWS SSO user** Authentication to the AWS SSO user portal is controlled by the directory that you have connected to AWS SSO. 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 SSO console. For more information, see Single Sign-On Access in the AWS Single Sign-On User Guide.
  - What level of permissions have been granted to the end users in the AWS SSO console to allow them the appropriate access to those AWS accounts. For more information, see Permission Sets in the AWS Single Sign-On 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. 237)
- Using Identity-Based Policies (IAM Policies) for AWS Control Tower (p. 240)

### 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 access a resource are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles), and 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 granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific actions that you want to allow on those resources.

**Topics**
In AWS Control Tower, the primary resource is a landing zone. AWS Control Tower also supports an additional resource type, guardrails. However, for AWS Control Tower, you can manage guardrails only in the context of an existing landing zone. Guardrails are referred to as a subresource.

Understanding 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 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.

Managing 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
- Identity-Based Policies (IAM Policies) (p. 238)
- Resource-Based Policies (p. 239)

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, the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:

  1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in Account A.
  2. Account A administrator attaches a trust policy to the role identifying Account B as the principal who can assume the role.
  3. Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A. The principal in the trust policy can also be an AWS service principal if you want to grant an AWS service permissions to assume the role.

For more information about using IAM to delegate permissions, see Access Management in the IAM User Guide.

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 (p. 74). 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. 240).

**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)

For more information about using identity-based policies with AWS Control Tower, see Using Identity-Based Policies (IAM Policies) for AWS Control Tower (p. 240). 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. 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 a IAM policy.

The following are the most basic policy elements:

• **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. 238).

• **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), the user that the policy is attached to 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](https://docs.aws.amazon.com/IAM/latest/UserGuide) 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](https://docs.aws.amazon.com/IAM/latest/UserGuide) in the *IAM User Guide*.

To express conditions, you 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](https://docs.aws.amazon.com/IAM/latest/UserGuide) 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](https://docs.aws.amazon.com/IAM/latest/UserGuide) (p. 74).

### 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](https://docs.aws.amazon.com/IAM/latest/UserGuide) (p. 237).
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. 241)
- AWSControlTowerServiceRolePolicy (p. 241)
- AWSControlTowerCloudTrailRole (p. 245)

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. 74).

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

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.

AWSControlTowerServiceRolePolicy

Managed Policy Name: AWSControlTowerServiceRolePolicy

The JSON artifact for AWSControlTowerServiceRolePolicy is the following:

```json
{
   "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"
         ]
      }
   ]
}
```
Using Identity-Based Policies (IAM Policies)

```json
{
    "Effect": "Allow",
    "Action": [
        "cloudformation:CreateStack",
        "cloudformation:CreateStackInstances",
        "cloudformation:CreateStackSet",
        "cloudformation:DeleteStack",
        "cloudformation:DeleteStackInstances",
        "cloudformation:DeleteStackSet",
        "cloudformation:DescribeStackInstance",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackSet",
        "cloudformation:DescribeStackSetOperation",
        "cloudformation:GetTemplate",
        "cloudformation:ListStackInstances",
        "cloudformation:UpdateStack",
        "cloudformation:UpdateStackInstances",
        "cloudformation:UpdateStackSet"
    ],
    "Resource": [
        "arn:aws:cloudformation::*:*:stack/AWSControlTower*/*",
        "arn:aws:cloudformation::*:*:stack/Set-AWSControlTower*/*",
        "arn:aws:cloudformation::*:*:stacket/AWSControlTower*:*",
        "arn:aws:cloudformation::*:*:stacket-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:logs::*:*:log-group:aws-controltower/CloudTrailLogs:*",
        "arn:aws:cloudtrail::*:*:trail/aws-controltower"
    ],
}
{
    "Effect": "Allow",
    "Action": ["s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::aws-controltower*/*"
    ],
}
{
    "Effect": "Allow",
    "Action": ["sts:AssumeRole"
    ],
    "Resource": [
        "arn:aws:iam::*:role/AWSControlTowerExecution"
    ],
}
{
    "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",
    "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",
"Resource": "*"
"Condition": {  
  "StringLike": {  
    "organizations:ServicePrincipal": "config.amazonaws.com"  
  }  
}  
}  
]  
}  
}  

Role trust policy:

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

The inline policy is **AWSControlTowerAdminPolicy**:

{  
  "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:

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

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "logs:CreateLogStream",
      "Effect": "Allow"
    },
    {
      "Action": "logs:PutLogEvents",
      "Effect": "Allow"
    }
  ]
}
```

**Managed policies for AWS Control Tower**

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AWSControlTowerServiceRolePolicy</strong></td>
<td>- Update to an existing policy  &lt;br&gt; AWS Control Tower added new permissions that allow customers to use KMS key encryption. &lt;br&gt; The KMS feature allows customers to provide their own KMS key to encrypt their AWS 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 <strong>AWSControlTowerAdmin</strong> 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 guardrails 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 Whitepaper** – 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.

AWS Control Tower is available in these AWS Regions:

- US East (N. Virginia)
- US East (Ohio)
- US West (Oregon)
- Canada (Central) Region
- Asia Pacific (Sydney)
- Asia Pacific (Singapore) Region
- Europe (Frankfurt) Region
- Europe (Ireland)
- Europe (London) Region
- Europe (Stockholm) Region
- Asia Pacific (Mumbai) Region
- Asia Pacific (Seoul) Region
- Asia Pacific (Tokyo) Region
- Europe (Paris) Region
- South America (São Paulo) Region
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. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 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. 277).
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 solution 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 guardrails 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

Logging of actions and events in AWS Control Tower is accomplished automatically through its integration with 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. Member account actions and events are viewable in 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. 249)
- Logging AWS Control Tower Actions with AWS CloudTrail (p. 249)
- Lifecycle Events in AWS Control Tower (p. 252)
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.

- **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.

- **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.

- **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.

For more information, see Logging AWS Control Tower Actions with AWS CloudTrail (p. 249).

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.

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.
AWS Control Tower User Guide
AWS Control Tower Information in CloudTrail

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
- 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:

- SetupLandingZone
- UpdateAccountFactoryConfig
- ManageOrganizationalUnit
- CreateManagedAccount
- EnableGuardrail
- GetLandingZoneStatus
- GetHomeRegion
- ListManagedAccounts
- DescribeManagedAccount
- DescribeAccountFactoryConfig
- DescribeGuardrailForTarget
- DescribeManagedOrganizationalUnit
- ListEnabledGuardrails
- ListGuardrailViolations
- ListGuardrails
- ListGuardrailsForTarget
- ListManagedAccountsForGuardrail
- ListManagedAccountsForParent
- ListManagedOrganizationalUnits
- ListManagedOrganizationalUnitsForGuardrail
- GetGuardrailComplianceStatus
- DescribeGuardrail
- ListDirectoryGroups
- DescribeSingleSignOn
- DescribeCoreService
- GetAvailableUpdates

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.

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 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",
      "creationDate": "2018-11-20T19:36:11Z"
    },
    "sessionIssuer": {
      "type": "Role",
      "principalId": "AIDACKCEVSQ6C2EXAMPLE",
      "arn": "arn:aws:iam::AKIAIOSFODNN7EXAMPLE:role/AWSControlTowerTestAdmin",
      "accountId": "AIDACKCEVSQ6C2EXAMPLE",
      "userName": "AWSControlTowerTestAdmin"
    }
  }
},
"eventTime": "2018-11-20T19:36:15Z",
"eventSource": "controltower.amazonaws.com",
"eventName": "SetupLandingZone",
"awsRegion": "us-east-1",
"sourceIPAddress": "AWS Internal",
"userAgent": "Coral/Netty4",
"errorCode": "InvalidParametersException",
"errorMessage": "Home region EU_CENTRAL_1 is unsupported",
"requestParameters": {
  "homeRegion": "EU_CENTRAL_1",
  "logAccountEmail": "HIDDEN_DUE_TO_SECURITY_REASONS",
  "sharedServiceAccountEmail": "HIDDEN_DUE_TO_SECURITY_REASONS",
  "securityAccountEmail": "HIDDEN_DUE_TO_SECURITY_REASONS",
  "securityNotificationEmail": "HIDDEN_DUE_TO_SECURITY_REASONS"
},
"responseElements": null,
"requestID": "96647b68-ed5f-4268-931c-807ced1f89e96",
"eventType": "AwsApiCall",
"recipientAccountId": "76543EXAMPLE"
}``
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 guardrails.

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 guardrail 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. 254): The log records whether AWS Control Tower successfully completed every action to create and provision a new account using account factory.
- **UpdateManagedAccount** (p. 254): 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. 255): The log records whether AWS Control Tower successfully completed every action to enable a guardrail on an OU that was created by AWS Control Tower.
- **DisableGuardrail** (p. 256): The log records whether AWS Control Tower successfully completed every action to disable a guardrail on an OU that was created by AWS Control Tower.
- **SetupLandingZone** (p. 257): The log records whether AWS Control Tower successfully completed every action to set up a landing zone.
- **UpdateLandingZone** (p. 258): The log records whether AWS Control Tower successfully completed every action to update your existing landing zone.
- **RegisterOrganizationalUnit** (p. 259): The log records whether AWS Control Tower successfully completed every action to enable its governance features on an OU.
- **DeregisterOrganizationalUnit** (p. 260): The log records whether AWS Control Tower successfully completed every action to disable its governance features on an OU.
- **PrecheckOrganizationalUnit** (p. 261): 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": "999cccaa-eeaa-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": "CreateManagedAccount",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "0000000-0000-0000-1111-123456789012",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "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.
EnableGuardrail

This lifecycle event records whether AWS Control Tower successfully enabled a guardrail 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 guardrailId and guardrailBehavior of the guardrail, and the organizationalUnitName and organizationalUnitId of the OU on which the guardrail is enabled.

```json
{
    "version": "0",
    "id": "999cccaea-eaaa-0000-1111-123456789012",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "XXXXXXXXXXXX",
    "region": "us-east-1",
    "resources": [],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "XXXXXXXXX",
            "invokedBy": "AWS Internal"
        },
        "eventTime": "2018-08-30T21:42:18Z",
        "eventSource": "controltower.amazonaws.com",
        "eventName": "UpdateManagedAccount",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "0000000000-0000-0000-1111-123456789012",
        "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"
        }
    }
}
```
DisableGuardrail

This lifecycle event records whether AWS Control Tower successfully disabled a guardrail 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 guardrail, and the organizationalUnitName and organizationalUnitId of the OU on which the guardrail is disabled.

```json
{
    "version": "0",
    "id": "999ccccaa-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": {
        "eventType": "EnableGuardrail",
        "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"
            }
        }
    }
}
```
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 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": "999ccaaa-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
    "resources": [ ],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": { // Management-account ID.
            "accountId": "XXXXXXXXXXXX",
            "invokedBy": "AWS Internal"
        },
```
"eventSource": "controltower.amazonaws.com",
"eventName": "SetupLandingZone",
"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": {
  "setupLandingZoneStatus": {
    "state": "SUCCEEDED",                         // Status of entire lifecycle operation.
    "message": "AWS Control Tower successfully set up a new 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",
    "completedTimestamp": "2018-08-30T21:42:18Z"
  }
}

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.

{  
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",                  // Request ID.
  "detail-type": "AWS Service Event via CloudTrail",
  "source": "aws.controltower",
  "account": "XXXXXXXXXXXX",
}
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.

```
"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": [  
            {  
                "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",
        "completedTimestamp": "2018-08-30T21:42:18Z"
    }
}
```
event. The lifecycle event log includes the organizationalUnitName and organizationalUnitId of the OU that AWS Control Tower has brought under its governance.

```json
{
    "version": "0",
    "id": "999cccaaeaaa-0000-1111-123456789012",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.controltower",
    "account": "123456789012",
    "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": "RegisterOrganizationalUnit",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "000000-0000-0000-1111-123456789012",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "registerOrganizationalUnitStatus": {
                "state": "SUCCEEDED",
                "message": "AWS Control Tower successfully registered an organizational unit."
            }
        }
    }
}
```

### 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 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": "999cccaaeaaa-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": "RegisterOrganizationalUnit",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "AWS Internal",
        "userAgent": "AWS Internal",
        "eventID": "000000-0000-0000-1111-123456789012",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "registerOrganizationalUnitStatus": {
                "state": "SUCCEEDED",
                "message": "AWS Control Tower successfully registered an organizational unit."
            }
        }
    }
}
```
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",  
  "eventDetails": {  
    "deregisterOrganizationalUnitStatus": {  
      "state": "SUCCEEDED",  
      "message": "AWS Control Tower successfully deregistered an organizational unit, and enabled mandatory guardrails on the new organizational unit."
    },  
    "organization": {  
      "organizationalUnitName": "Test",  
      "organizationalUnitId": "ou-adpf-302pk332"  
    }  
  }  
}
```
"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 pre-check 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"
},
"eventCategory": "Management"
Walkthroughs

This chapter contains walkthrough procedures that can help you in your use of AWS Control Tower.

Topics

- Walkthrough: Clean up AWS Control Tower Resources (p. 263)
- Walkthrough: Decommission an AWS Control Tower Landing Zone (p. 268)
- Walkthrough: Configure AWS Control Tower Without a VPC (p. 272)
- Walkthrough: Automate Account Provisioning in AWS Control Tower by AWS Service Catalog APIs (p. 274)
- Walkthrough: Set Up Security Groups in AWS Control Tower With AWS Firewall Manager (p. 277)
- Walkthrough: Move from ALZ to AWS Control Tower (p. 277)

Walkthrough: Clean up AWS Control Tower Resources

This document provides instructions for how to delete 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 is not the same as decommissioning your landing zone.

**Warning**
Manually deleting 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. 268) 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.

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. 269).

**Warning**
We strongly recommend that you perform this decommissioning process only if you intend to stop using your landing zone. This process cannot be undone.
About manual cleanup of AWS Control Tower resources

The individual procedures in this chapter guide you through manual methods of cleaning up AWS Control Tower resources. These procedures can be followed any time you need to delete specific resources from your landing zone. Two types of tasks may require cleanup of resources:

- To delete resources as you manage your landing zone in ordinary situations.
- To clean up resources that remain after automated decommissioning.

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 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. 264)
- Delete StackSets and Stacks (p. 264)
- Delete Amazon S3 Buckets in the Log Archive Account (p. 265)
- Clean Up Account Factory (p. 266)
- Clean Up Roles and Policies (p. 267)
- AWS Control Tower cleanup help (p. 268)

**Delete SCPs**

AWS Control Tower uses service control policies (SCPs) for its guardrails. This procedure walks through how to delete the SCPs specifically related to AWS Control Tower.

**To delete AWS Organizations SCPs**

1. Open the Organizations console at https://console.aws.amazon.com/organizations/.
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 guardrails 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 AWS SSO user in the **AWSControlTowerExecution** group and then delete the Amazon S3 buckets in your log archive account.

To sign in to your log archive execution 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.

Clean Up Account Factory

The following procedure guides you through how to sign in as an AWS SSO user in the AWSServiceCatalogAdmins group and then clean up your Account Factory accounts.

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 AWS 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.
Clean Up Roles and Policies

These procedures walk you through how to clean up the roles and policies that were created when your landing zone was set up.

**To delete the AWS SSO AWSServiceCatalogEndUserAccess role**

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**

1. Open the IAM console at https://console.aws.amazon.com/iam/.
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**

1. Open the IAM console at https://console.aws.amazon.com/iam/.
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 cleanup help**

If you encounter any issues that you can’t resolve during the clean-up process, contact [AWS Support](#).

---

**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. 268).

**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 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 guardrail enabled in the landing zone. AWS Control Tower deletes the AWS CloudFormation resources supporting the guardrail.
• Disables each preventive guardrail 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 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 Single Sign-On (SSO)

For customers without an existing AWS SSO directory, AWS Control Tower sets up AWS SSO and configures an initial directory. When you decommission your landing zone, AWS Control Tower makes no changes to AWS SSO. If needed, you can delete the AWS SSO 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 AWS SSO permission sets are not removed.
• AWS SSO 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.

CloudWatch Logs Log Group
A CloudWatch Logs log group, aws-controltower/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.

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 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 manual cleanup of AWS Control Tower resources (p. 264).

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:

```
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 logging and audit accounts if you create a new landing zone after decommissioning one.
- The CloudWatch Logs log group, `aws-controltower/CloudTrailLogs`, must be deleted manually before you set up another landing zone.
- The two 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 Sign in as a Root User (p. 30) to the audit account and to the logging account and delete them individually.

- You may wish to delete the AWS Single Sign-On (AWS SSO) configuration for AWS Control Tower manually, but you can proceed with the existing AWS SSO configuration.
- You may wish to remove the VPC created by AWS Control Tower, and remove the associated AWS CloudFormation stack set.
- Before you can set up an AWS Control Tower landing zone in a different home Region, you also must run the command `aws organizations disable-aws-service-access --service-principal controltower.amazonaws.com`.

**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:
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 guardrail Disallow internet access for an Amazon VPC instance managed by a customer (p. 211). 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/. 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/.
2. Search for VPC or select VPC from the AWS 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 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.
      - Change the **Address range (CIDR) restriction for account VPCs** value to `10.0.0.0/16`
   
   b. **Option 2 – Removing AWS Regions**
      - Clear every checkbox in the **Regions for VPC creation** column.
      - Change the **Address range (CIDR) restriction for account VPCs** value to `10.0.0.0/16`

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: Automate Account Provisioning in AWS Control Tower by AWS 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. 124).

**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.

```yaml
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.

```yaml
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

Resources:
  # This needs to run after AdminRoleName exists.
  ExecutionRole:
    Type: "AWS::IAM::Role"
    Properties:
      RoleName: !Ref ExecutionRoleName
```

275
MaxSessionDuration: !Ref SessionDurationInSecs
AssumeRolePolicyDocument:
  Version: "2012-10-17"
  Statement:
  - Effect: "Allow"
    Principal:
      AWS:
      - !Sub "arn:aws:iam::${AdminAccountId}:role/${AdminRoleName}"
    Action:
    - "sts:AssumeRole"
    Path: "/"
  ManagedPolicyArns:
  - "arn:aws:iam::aws:policy/AdministratorAccess"

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: "lpv2-7n2o3nudljh4e",
  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: 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 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: 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.
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.

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

- US East (N. Virginia)
- US East (Ohio)
- US West (Oregon)
- Canada (Central) Region
- Asia Pacific (Sydney)
- Asia Pacific (Singapore) Region
- Europe (Frankfurt) Region
- Europe (Ireland)
- Europe (London) Region
- Europe (Stockholm) Region
- Asia Pacific (Mumbai) Region
- Asia Pacific (Seoul) Region
- Asia Pacific (Tokyo) Region
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 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 Account Factory accounts with AWS Service Catalog](p. 118).

**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 guardrail 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 > AWS StackSets](aws-control-tower-stacksets)
- Check each stack set related to AWS Control Tower (prefix: "AWSControlTower")
- Look for AWS 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 guardrails 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 AWS 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 AWS 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 guardrails 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. If the AWS Config service is enabled in any one of the Regions supported by AWS Control Tower, it can result in this failure.

**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-dpqgywxxxx

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.
- Your SSO user has not been added to the appropriate permission group. You may need to add your SSO 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.

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 SSO 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 guardrails are not taking effect on accounts

If you've recently expanded your AWS Control Tower deployment into a new AWS Region, newly-applied detective guardrails 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 guardrails on existing accounts are still in effect.

If you try to enable a detective guardrail before updating your accounts, you may see an error message similar to this one:

AWS Control Tower can't enable the selected guardrail on this OU. AWS Control Tower cannot apply the guardrail on the OU ou-xxx-xxxxxx, 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 Update existing OUs and accounts (p. 169).

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. 277). You can substitute the UpdateProvisionedProduct API for the ProvisionProduct API shown in the video.

If you have further difficulties with enabling detective guardrails 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-dXXXXXXee31 has reached SUCCEEDED state but has 1 NON-CURRENT stack instances; here is the summary: {
StackSet Id: AWSControlTowerBP-BASELINE-CLOUDWATCH:40XXbf2-Xead-46a1-XXXa-eXXXecb2ee2,
Stack instance Id: arn:aws:cloudformation:eu-west-1:1X23456789XX:stack/StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-4feXXXXX-ecXX-XXc6-bXXX-4e6878/4feXXXXX-ecX-4ae123458/4feXXXXX-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 AWS SSO and AWS Control Tower.
• Manage access to AWS centrally for Okta users with AWS Single Sign-On – This blog post describes how to use Okta with AWS SSO 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 guardrails and how to subscribe to SNS notifications so that you can be notified by email of guardrail 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 SSO, 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. 304).

Features released in 2022 (p. 289)
Features released in 2021 (p. 289)
Features released in 2020 (p. 297)
Features released in 2019 (p. 301)

January 2022 - Present

Since January 2022, AWS Control Tower has released the following updates:

- AWS Control Tower landing zone version 2.8 (p. 290)

January - December 2021

Since January 2021, AWS Control Tower has released the following updates:

- Region deny capabilities (p. 290)
- Data residency features (p. 291)
- AWS Control Tower introduces Terraform account provisioning and customization (p. 291)
- New lifecycle event available (p. 291)
- AWS Control Tower enables nested OUs (p. 291)
- Detective guardrail concurrency (p. 292)
- Two new Regions available (p. 293)
- Region deselection (p. 293)
- AWS Control Tower works with AWS Key Management Systems (p. 293)
- Guardrails renamed, functionality unchanged (p. 294)
- AWS Control Tower scans SCPs daily to check for drift (p. 294)
- Customized names for OUs and accounts (p. 294)
- AWS Control Tower landing zone version 2.7 (p. 295)
- Three new AWS Regions available (p. 296)
- Govern selected Regions only (p. 296)
- AWS Control Tower now extends governance to existing OUs in your AWS organizations (p. 296)
- AWS Control Tower provides bulk account updates (p. 296)
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. 36))

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 guardrail, 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. 35).

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 guardrail (p. 101).
Data residency features

November 30, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower now offers purpose-built guardrails 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 guardrails, 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 guardrails in the AWS Control Tower console. For a full list of available guardrails, see Guardrail reference (p. 181).

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. 126), 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. 261).

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 guardrails hierarchically. You can register OUs containing nested OUs, create and register OUs under parent OUs, and enable guardrails 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 guardrails on multiple OUs, by enforcing guardrails 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. 164).
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 guardrails (SCPs) are inherited down the hierarchy automatically; SCPs applied to the parent are inherited by all nested OUs.
4. Detective guardrails (AWS Config rules) are NOT inherited automatically.
5. Compliance with detective guardrails 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 guardrail concurrency**

November 5, 2021

(Optional update for AWS Control Tower landing zone)

AWS Control Tower detective guardrails now support concurrent operations for detective guardrails, improving the ease of use and performance. You can enable multiple detective guardrails without waiting for individual guardrail operations to complete.

**Supported functionality:**

- Enable different detective guardrails 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 guardrails on different OUs, concurrently.
- Guardrail error messaging has been improved to give additional guidance for supported guardrail concurrency operations.

**Not supported:**

- Enabling the same detective guardrail on multiple OUs concurrently is not supported.
- Preventive guardrail concurrency is not supported.

You can experience the detective guardrail 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 Tower deploys its governance capabilities—which include centralized logging, monitoring, and guardrails — in your chosen Regions in the accounts. Choosing to deselect a Region and remove AWS Control Tower governance from that Region removes that governance functionality, but it does not inhibit your users' ability to deploy AWS resources or workloads into those Regions.

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 AWS 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.

Guardrails renamed, functionality unchanged

July 26, 2021

(No update required for AWS Control Tower landing zone)

AWS Control Tower is revising certain guardrails names and descriptions to better reflect the policy intentions of the guardrail. The revised names and descriptions help you understand more intuitively the ways in which guardrails enhance control of your accounts. For example, we changed part of the names of detective guardrails from “Disallow” to “Detect” because the detective guardrail itself does not stop a specific action, it only detects policy violations and provides alerts through the dashboard.

Guardrail functionality, guidance, and implementation remain unchanged. Only the guardrail 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 guardrails are applied correctly and that they have not drifted. If a 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.

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 set up 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. 36))

With AWS Control Tower version 2.7, AWS Control Tower introduces four new mandatory preventative Log Archive guardrails that implement policy solely on AWS Control Tower resources. We have adjusted the guidance on four existing Log Archive guardrails from mandatory to elective, because they set policy for resources outside of AWS Control Tower. This guardrail 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 guardrails can be used in conjunction with the new mandatory guardrails to provide governance to a broader set of AWS Log Archives. Existing AWS Control Tower environments will keep these four changed guardrails enabled automatically, for environment consistency; however, these elective guardrails now can be disabled. New AWS Control Tower environments must enable all elective guardrails. Existing environments must disable the formerly mandatory guardrails before adding encryption to S3 buckets that are not deployed by AWS Control Tower.

New mandatory guardrails:

• 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 guardrails in your chosen Regions in the accounts. For more information about selecting Regions, see Configure your AWS Control Tower Regions (p. 99).

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. 167).

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. 170).

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 guardrails 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. 36).

For information about registering or re-registering an OU, see Register an existing organizational unit with AWS Control Tower (p. 167).

January - December 2020

Since January 1, 2020, AWS Control Tower has released the following updates:

- AWS Control Tower console now links to external AWS Config rules (p. 297)
- AWS Control Tower now available in additional Regions (p. 298)
- Guardrail update (p. 298)
- AWS Control Tower console shows more detail about OUs and accounts (p. 298)
- Use AWS Control Tower to set up new multi-account AWS environments in AWS Organizations (p. 299)
- Customizations for AWS Control Tower solution (p. 299)
- General availability of AWS Control Tower version 2.3 (p. 299)
- Single-step account provisioning in AWS Control Tower (p. 300)
- AWS Control Tower decommissioning tool (p. 300)
- AWS Control Tower lifecycle event notifications (p. 301)

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. 36))

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 guardrails applied to your accounts so you can track compliance and determine if you need additional guardrails for your account. For information, see How AWS Control Tower aggregates AWS Config rules in unmanaged OUs and accounts (p. 93).
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. 36))

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. 99).

Guardrail update

October 8, 2020

(No update required for AWS Control Tower landing zone)

An updated version has been released for the mandatory guardrail AWS-GR_IAM_ROLE_CHANGE_PROHIBITED.

This change to the guardrail is required because accounts that are being enrolled automatically into AWS Control Tower must have the AWSControlTowerExecution role enabled. The previous version of the guardrail prevents this role from being created.

For more information, see Guardrail update (p. 191) in the AWS Control Tower User Guide Guardrail 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. 10). 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 Create or Enroll An Individual Account (p. 118).

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. 252)) 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. 36).)
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 guardrails as a result of a deployment to a new Region, see Configure your AWS Control Tower Regions (p. 99).

### 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 provisioned account and to the single sign-on (SSO) user that is created for the account. If you require these two email addresses to be different, you must provision your account through AWS Service Catalog.

Update accounts that you create through quick account provisioning by using AWS Service Catalog and the AWS Control Tower account factory, just like updates to any other account.

**Note**
In April 2020, the **Quick account provisioning** capability was renamed to **Enroll account**. It now permits enrollment of existing AWS accounts as well as creation of new accounts. For more information, see Create or Enroll An Individual Account (p. 118).

### 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. 268).
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. 252) marks the completion of an AWS Control Tower action that can change the state of resources such as organizational units (OUs), accounts, and guardrails 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 guardrail 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. 301)
- New elective guardrails in AWS Control Tower (p. 302)
- New detective guardrails in AWS Control Tower (p. 302)
- AWS Control Tower accepts email addresses for shared accounts with different domains than the management account (p. 302)
- General availability of AWS Control Tower version 2.1 (p. 303)

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. 36).)

AWS Control Tower version 2.2 provides three new preventive guardrails that prevent drift in accounts:

- Disallow Changes to Amazon CloudWatch Logs Log Groups set up by AWS Control Tower (p. 184)
- Disallow Deletion of AWS Config Aggregation Authorizations Created by AWS Control Tower (p. 185)
- Disallow Deletion of Log Archive (p. 185)

A guardrail 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 guardrails. 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. 148).

New elective guardrails 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 guardrails:

- Disallow Delete Actions on Amazon S3 Buckets Without MFA (p. 207)
- Disallow Changes to Replication Configuration for Amazon S3 Buckets (p. 206)
- Disallow Actions as a Root User (p. 196)
- Disallow Creation of Access Keys for the Root User (p. 195)

A guardrail 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 Guardrails in AWS Control Tower (p. 173).

New detective guardrails 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 guardrails:

- Detect Whether Versioning for Amazon S3 Buckets is Enabled (p. 209)
- Detect Whether MFA is Enabled for AWS IAM Users of the AWS Console (p. 208)
- Detect Whether MFA is Enabled for AWS IAM Users (p. 207)
- Detect Whether Amazon EBS Optimization is Enabled for Amazon EC2 Instances (p. 202)
- Detect Whether Amazon EBS Volumes are Attached to Amazon EC2 Instances (p. 201)
- Detect Whether Public Access to Amazon RDS Database Instances is Enabled (p. 202)
- Detect Whether Public Access to Amazon RDS Database Snapshots is Enabled (p. 203)
- Detect Whether Storage Encryption is Enabled for Amazon RDS Database Instances (p. 203)

A guardrail is a high-level rule that provides ongoing governance for your overall AWS environment. A detective guardrail detects noncompliance of resources within your accounts, such as policy violations, and provides alerts through the dashboard. For more information, see Guardrails in AWS Control Tower (p. 173).

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. 36).)

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. 19) that employs best practices such as configuring a multi-account structure using AWS Organizations, managing user identities and federated access with AWS Single Sign-On, enabling account provisioning through AWS Service Catalog, and creating a centralized log archive using AWS CloudTrail and AWS Config.

For ongoing governance, you can enable pre-configured guardrails, 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, guardrails 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 guardrails.
Document history

- **Latest documentation update:** February 10, 2022

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>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updated support for AWS best practices, version 2.8</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 guardrail</td>
<td>AWS Control Tower now includes a guardrail 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 guardrails</td>
<td>AWS Control Tower now support guardrails 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</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</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</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 guardrail concurrency</td>
<td>AWS Control Tower detective guardrails now support concurrent enable and disable operations.</td>
<td>November 5, 2021</td>
</tr>
<tr>
<td>Two new regions available</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>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Region deselection (p. 304)</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>KMS keys available (p. 304)</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>Change to a managed policy (p. 304)</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>Guardrail names changed, functionality unchanged (p. 304)</td>
<td>Certain guardrail names and descriptions were updated to better reflect the policy intentions of the guardrail, with no change in functionality.</td>
<td>July 26, 2021</td>
</tr>
<tr>
<td>Automated scans of managed SCPs (p. 304)</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>Customized names for OUs and accounts (p. 304)</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>Decommissioning a landing zone is self-service (p. 304)</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>Three additional Regions (p. 304)</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 guardrails, landing zone version 2.7 available (p. 304)</strong></td>
<td>Four new Log Archive guardrails provide Log Archive governance over AWS Control Tower resources, separately from governance of resources outside of AWS Control Tower. Guidance on four existing guardrails 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. 304)</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. 304)</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. 304)</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. 304)</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>Event Description</td>
<td>Event Details</td>
<td>Date</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>AWS Control Tower is available in more AWS Regions. (p. 304)</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>Guardrail update (p. 304)</td>
<td>An updated version has been released for the mandatory guardrail <strong>AWS-GR_IAM_ROLE_CHANGE_PROHIBITED</strong>. The updated guardrail 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. 304)</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. 304)</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. 304)</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 AWSAWSaccounts as well as creation of new accounts.</td>
<td>April 16, 2020</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td><strong>AWS Control Tower is now available in Asia Pacific</strong> (p. 304)</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><strong>Decommissioning an AWS Control Tower landing zone is possible</strong> (p. 304)</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><strong>Quick account provisioning is available in AWS Control Tower</strong> (p. 304)</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><strong>Lifecycle events are tracked in AWS Control Tower</strong> (p. 304)</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><strong>Settings and Activities pages are available for AWS Control Tower</strong> (p. 304)</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><strong>Additional preventive guardrails are available for AWS Control Tower</strong> (p. 304)</td>
<td>Preventive guardrails in AWS Control Tower keep your organization and resources aligned with your environment.</td>
<td>September 6, 2019</td>
</tr>
<tr>
<td><strong>Additional detective guardrails are available for AWS Control Tower</strong> (p. 304)</td>
<td>Detective guardrails in AWS Control Tower give information about the state of your organization and resources.</td>
<td>August 27, 2019</td>
</tr>
<tr>
<td><strong>AWS Control Tower is now generally available</strong> (p. 304)</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>
AWS glossary

For the latest AWS terminology, see the AWS glossary in the AWS General Reference.