Implementation Guide

# Automated Security Response on AWS

# Automated Security Response on AWS: Implementation Guide

# Table of Contents

# Automatically address security threats with predefined response and remediation actions in AWS Security Hub

This implementation guide provides an overview of the Automated Security Response on AWS solution, its reference architecture and components, considerations for planning the deployment, configuration steps for deploying the Automated Security Response on AWS solution to the Amazon Web Services (AWS) Cloud.

Use this navigation table to quickly find answers to these questions:

| If you want to . . . | Read . . . |
|---|---|
| Know the cost for running this solution | Cost |
| Understand the security considerations for this solution | Security |
| Know how to plan for quotas for this solution | Quotas |
| Know which AWS Regions are supported for this solution | Supported AWS Regions |
| View or download the AWS CloudFormation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this solution | AWS CloudFormation templates |
| Access the source code and optionally use the AWS Cloud Development Kit (AWS CDK) to deploy the solution. | GitHub repository |

The continued evolution of security requires proactive steps to secure data which can make it difficult, expensive, and time-consuming for security teams to react. The Automated Security Response on AWS solution helps you quickly react to address security issues by providing predefined responses and remediation actions based on industry compliance standards and best practices.

Automated Security Response on AWS is an AWS Solution that works with AWS Security Hub to improve your security and helps align your workloads to the Well-Architected Security pillar best practices (SEC10). This solution makes it easier for AWS Security Hub customers to resolve common security findings and improve their security posture in AWS.

You can select specific playbooks to deploy in your Security Hub primary account. Each playbook contains the necessary custom actions, Identity and Access Management (IAM) roles, Amazon EventBridge rules, AWS Systems Manager automation documents, AWS Lambda functions, and AWS Step Functions needed to start a remediation workflow within a single AWS account, or across multiple accounts. Remediations work from the Actions menu in AWS Security Hub and allow authorized users to remediate a finding across all of their AWS Security Hub-managed accounts with a single action. For example, you can apply recommendations from the Center for Internet Security (CIS) AWS Foundations Benchmark, a compliance standard for securing AWS resources, to ensure passwords expire within 90 days and enforce encryption of event logs stored in AWS.

**Note**

Remediation is intended for emergent situations that require immediate action. This solution makes changes to remediate findings only when initiated by you via the AWS Security Hub Management console, or when automated remediation has been enabled using the Amazon EventBridge rule for a specific control. To revert these changes, you must manually put resources back in their original state.
When remediating AWS resources deployed as a part of the CloudFormation stack, be aware that this might cause a drift. When possible, remediate stack resources by modifying the code that defines the stack resources and updating the stack. For more information, refer to What is drift? in the *AWS CloudFormation User Guide*.

Automated Security Response on AWS includes the playbook remediations for the security standards defined as part of the following:

- Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0
- CIS AWS Foundations Benchmark v1.4.0
- CIS AWS Foundations Benchmark v3.0.0
- AWS Foundational Security Best Practices (FSBP) v.1.0.0
- Payment Card Industry Data Security Standard (PCI-DSS) v3.2.1
- National Institute of Standards and Technology (NIST) SP 800-53 Rev. 5

The solution also includes a Security Controls (SC) playbook for the consolidated control findings feature of AWS Security Hub. For more information, refer to Playbooks.

This implementation guide discusses architectural considerations and configuration steps for deploying the Automated Security Response on AWS solution in the AWS Cloud. It includes links to AWS CloudFormation templates that launch, configure, and run the AWS compute, network, storage, and other services required to deploy this solution on AWS, using AWS best practices for security and availability.

The guide is intended for IT infrastructure architects, administrators, and DevOps professionals who have practical experience architecting in the AWS Cloud.

# Features and benefits

The Automated Security Response on AWS provides the following features:

**Automatically remediate findings for specific controls**

Activate Amazon EventBridge rules for controls to automatically remediate findings for that control immediately after they appear in AWS Security Hub.

**Manage remediations across multiple accounts and Regions from one location**

From an AWS Security Hub administrator account that is configured as the aggregation destination for your organization's accounts and Regions, initiate a remediation for a finding in any account and Region in which the solution is deployed.

**Get notified of remediation actions and results**

Subscribe to the Amazon SNS topic deployed by the solution to be notified when remediations are initiated and whether or not the remediation was successful.

**Integrate with ticket systems like Jira or ServiceNow**

To help your organization react to remediations (for example, updating your infrastructure code), this solution can push tickets to your external ticketing system.

**Use AWSConfigRemediations in the GovCloud and China partitions**

Some of the remediations included in the solution are repackages of AWS-owned AWSConfigRemediation documents that are available in the commercial partition but not in GovCloud or China. Deploy this solution to make use of these documents in those partitions.

### Extend the solution with custom remediation and Playbook implementations

The solution is designed to be extensible and customizable. To specify an alternative remediation implementation, deploy customized AWS Systems Manager automation documents and AWS IAM Roles. To support an entire new set of controls that is not implemented by the solution, deploy a custom Playbook.

## Use cases

### Enforce compliance to a standard across your organization's accounts and Regions

Deploy the Playbook for a standard (for example, AWS Foundational Security Best Practices) to be able to use the provided remediations. Automatically or manually initiate remediations for resources in any account and Region in which the solution is deployed to fix resources that are out of compliance.

### Deploy custom remediations or Playbooks to meet your organization's compliance needs

Use the provided Orchestrator components as a framework. Build custom remediations to address out-of-compliance resources according to your organization's specific needs.

## Concepts and definitions

This section describes key concepts and defines terminology specific to this solution:

### remediation, remediation runbook

An implementation of a set of steps that resolves a finding. For example, a remediation for the control Security Control (SC) Lambda.1 "Lambda function policies should prohibit public access" would modify the policy of the relevant AWS Lambda Function to remove statements that allow public access.

### control runbook

One of a set of AWS Systems Manager (SSM) automation documents that the Orchestrator uses to route an initiated remediation for a specific control to the correct remediation runbook. For example, the remediations for SC Lambda.1 and AWS Foundational Security Best Practices (FSBP) Lambda.1 are implemented with the same remediation runbook. The Orchestrator invokes the control runbook for each control, which are named ASR-AFSBP_Lambda.1 and ASR-

SC_2.0.0_Lambda.1, respectively. Each control runbook invokes the same remediation runbook, which in this case would be ASR-RemoveLambdaPublicAccess.

**orchestrator**

The Step Functions deployed by the solution that takes as input a finding object from AWS Security Hub and invokes the correct control runbook in the target account and Region. The Orchestrator also notifies the solution SNS Topic when the remediation is started and when the remediation succeeds or fails.

**standard**

A group of controls defined by an organization as part of a compliance framework. For example, one of the standards supported by AWS Security Hub and this solution is AWS FSBP.

**control**

A description of the properties that a resource should or should not have in order to be in compliance. For example, the control AWS FSBP Lambda.1 states that AWS Lambda Functions should prohibit public access. A function that allows public access would fail this control.

**consolidated control findings, security control, security controls view**

A feature of AWS Security Hub that, when activated, displays findings with their consolidated control IDs rather than IDs that correspond to a particular standard. For example, the controls AWS FSBP S3.2, CIS v1.2.0 2.3, CIS v1.4.0 2.1.5.2, and PCI-DSS v3.2.1 S3.1 all map to the consolidated (SC) control S3.2 "S3 Buckets should prohibit public read access." When this feature is turned on, SC runbooks are used.

For a general reference of AWS terms, refer to the AWS Glossary.

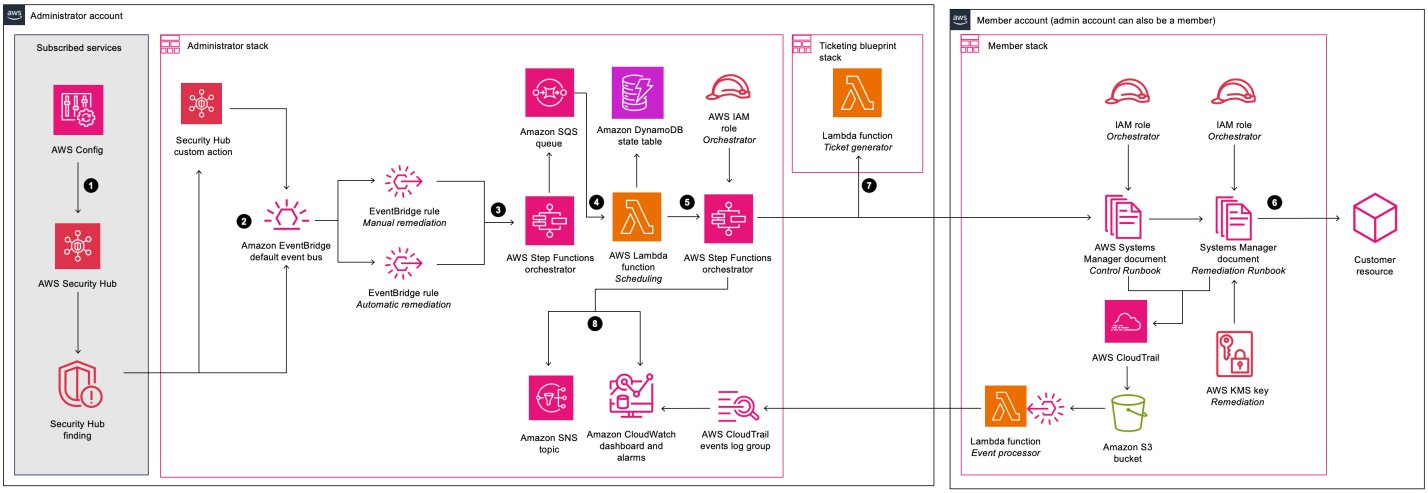
# Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this solution.

# Architecture diagram

Deploying this solution with the default parameters builds the following environment in the AWS Cloud.

### Automated Security Response on AWS architecture




> **Note**
>
> AWS CloudFormation resources are created from AWS Cloud Development Kit (AWS CDK) constructs.

The high-level process flow for the solution components deployed with the AWS CloudFormation template is as follows:

1. **Detect**: AWS Security Hub provides customers with a comprehensive view of their AWS security state. It helps them to measure their environment against security industry standards and best practices. It works by collecting events and data from other AWS services, such as AWS Config, Amazon Guard Duty, and AWS Firewall Manager. These events and data are analyzed against

security standards, such as CIS AWS Foundations Benchmark. Exceptions are asserted as *findings* in the AWS Security Hub console. New findings are sent as Amazon EventBridge events.

2. **Initiate**: You can initiate events against findings using custom actions, which result in EventBridge events. AWS Security Hub custom actions and EventBridge rules initiate Automated Security Response on AWS playbooks to address findings. The solution deploys:

   a. One EventBridge rule to match the custom action event

   b. One EventBridge event rule for each supported control (deactivated by default) to match the real-time finding event

   You can use the **Custom actions** menu in the Security Hub console to initiate automated remediation. After careful testing in a non-production environment, you can also activate automated remediations. You can activate automations for individual remediations — you don't need to activate automatic initiations on all remediations.

3. **Pre-remediate**: In the admin account, AWS Step Functions processes the remediation event and prepares it to be scheduled.

4. **Schedule**: The solution invokes the scheduling AWS Lambda function to place the remediation event in the Amazon DynamoDB state table.

5. **Orchestrate**: In the admin account, Step Functions uses cross-account AWS Identity and Access Management (IAM) roles. Step Functions invokes the remediation in the member account containing the resource that produced the security finding.

6. **Remediate**: An AWS Systems Manager Automation document in the member account performs the action required to remediate the finding on the target resource, such as disabling Lambda public access.

   Optionally, you can enable the Action Log feature in the member stacks with the **EnableCloudTrailForASRActionLog** parameter. This feature captures actions taken by the solution in your Member accounts and displays them in the solution's Amazon CloudWatch dashboard.

7. **(Optional) Create a ticket**: If you use the **TicketGenFunctionName** parameter to enable ticketing in the Admin stack, the solution invokes the provided ticket generator Lambda function. This Lambda function creates a ticket in your ticketing service after the remediation has successfully executed in the Member account. We provide stacks for integration with Jira and ServiceNow.

8. **Notify and log**: The playbook logs the results to a CloudWatch log group, sends a notification to an Amazon Simple Notification Service (Amazon SNS) topic, and updates the Security Hub finding. The solution maintains an audit trail of actions in the finding notes.

# AWS Well-Architected design considerations

This solution was designed with best practices from the AWS Well-Architected Framework which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud. This section describes how the design principles and best practices of the Well-Architected Framework were applied when building this solution.

## Operational excellence

This section describes how we architected this solution using the principles and best practices of the operational excellence pillar.

- Resources defined as IaC using CloudFormation.
- Remediations implemented with the following characteristics, where possible:
  - Idempotency
  - Error handling and reporting
  - Logging
  - Restoring resources to a known state on failure

## Security

This section describes how we architected this solution using the principles and best practices of the security pillar.

- IAM used for authentication and authorization.
- Role permissions scoped to be as narrow as possible, though in many cases this soloution requires wildcard permissions to be able to act on any resources.

## Reliability

This section describes how we architected this solution using the principles and best practices of the reliability pillar.

- Security Hub continues to create findings if the underlying cause of the finding is not resolved by the remediation.
- Serverless services allow the solution to scale as needed.

## Performance efficiency

This section describes how we architected this solution using the principles and best practices of the performance efficiency pillar.

- This solution was designed to be a platform for you to extend without having to implement orchestration and permissions yourself.

## Cost optimization

This section describes how we architected this solution using the principles and best practices of the cost optimization pillar.

- Serverless services allow you to pay for only what you use.
- Use the free tier for SSM automation in every account

## Sustainability

This section describes how we architected this solution using the principles and best practices of the sustainability pillar.

- Serverless services allow you to scale up or down as needed.

# Architecture details

This section describes the components and AWS services that make up this solution and the architecture details on how these components work together.

## AWS Security Hub integration

Deploying the `automated-security-response-admin` stack creates integration with AWS Security Hub's custom action feature. When AWS Security Hub console users select **Findings for remediation**, the solution routes the finding record for remediation using an AWS Step Functions.

Cross-account permissions and AWS Systems Manager runbooks must be deployed to all AWS Security Hub accounts (admin and member) using the `automated-security-response-member.template` and `automated-security-response-member-roles.template` CloudFormation templates. For more information, refer to Playbooks. This template allows automated remediation in the target account.

Users can automatically initiate automated remediations on a per-remediation basis using Amazon CloudWatch events rules. This option activates fully automatic remediation of findings as soon as they are reported to AWS Security Hub. By default, automatic initiations are turned off. This option can be changed at any time during or after installation of the playbook by turning on the CloudWatch Events rules in the AWS Security Hub admin account.

## Cross-account remediation

Automated Security Response on AWS uses cross-account roles to work across primary and secondary accounts using cross-account roles. These roles are deployed to member accounts during solution installation. Each remediation is assigned an individual role. The remediation process in the primary account is granted permission to assume the remediation role in the account that requires remediation. Remediation is performed by AWS Systems Manager runbooks running in the account that requires remediation.

## Playbooks

A set of remediations is grouped into a package called a *playbook*. Playbooks are installed, updated, and removed using this solution's templates. For information about supported remediations

in each playbook, refer to Developer Guide → Playbooks. This solution currently supports the following playbooks:

- Security Control, a playbook aligned with the Consolidated control findings feature of AWS Security Hub, published February 23, 2023.

> **Important**
>
> When Consolidated control findings are enabled in Security Hub, this is the only playbook that should be enabled in the solution.

- Center for Internet Security (CIS) Amazon Web Services Foundations benchmarks, version 1.2.0, published May 18, 2018.
- Center for Internet Security (CIS) Amazon Web Services Foundations benchmarks, version 1.4.0, published November 9, 2022.
- Center for Internet Security (CIS) Amazon Web Services Foundations benchmarks, version 3.0.0, published May 13, 2024.
- AWS Foundational Security Best Practices (FSBP) version 1.0.0, published March 2021.
- Payment Card Industry Data Security Standards (PCI-DSS) version 3.2.1, published May 2018.
- National Institute of Standards and Technology (NIST) version 5.0.0, published November 2023.

### Centralized logging

Automated Security Response on AWS logs to a single CloudWatch Logs group, SO0111-ASR. These logs contain detailed logging from the solution for troubleshooting and management of the solution.

## Notifications

This solution uses an Amazon Simple Notification Service (Amazon SNS) topic to publish remediation results. You can use subscriptions to this topic to extend the capabilities of the solution. For example, you can send email notifications and update trouble tickets.

- **SO0111-ASR_Topic** – Used to send general informational and error messages related to executed remediations.

- **SO0111-ASR_Alarm_Topic** – Used to notify when one of the solution's alarms is triggered, indicating that the solution is not functioning as expected.

## AWS services in this solution

The solution uses the following services. Core services are required to use the solution, and supporting services connect the core services.

| AWS service | Description |
|---|---|
| Amazon EventBridge | **Core**. Deploys events that will initiate the orchestator step function when a finding is being remediated. |
| AWS IAM | **Core**. Deploys many roles to allow remediati ons on different resources. |
| AWS Lambda | **Core.** Deploys multiple lambda functions that will be used by the step function orchestator to remediate issues. |
| AWS Security Hub | **Core**. Provides customers with a comprehen sive view of their AWS security state. |
| AWS Step Functions | **Core**. Deploys an orchestrator that will invoke the remediation documents with AWS Systems Manager API calls. |
| AWS Systems Manager | **Core**. Deploys System Manager Documents (link to doc) that contain the remediation logic that will be ran. |
| AWS CloudTrail | **Supporting.** Records changes that the solution makes to your AWS resources and displays them on a CloudWatch dashboard. |
| Amazon CloudWatch | **Supporting**. Deploys log groups that the different playbooks will use to log results. |

| AWS service | Description |
| --- | --- |
| | Collects metrics to display on a custom dashboard with alarms. |
| AWS DynamoDB | **Supporting**. Stores the last run remediation in each account and Region to optimize scheduling of remediations. |
| Amazon Simple Notification Service | **Supporting**. Deploys SNS topics that receive a notification once a remediation has been completed. |
| AWS SQS | **Supporting**. Assists with scheduling remediations so that the solution can run remediations in parallel. |
| AWS Key Management Service | **Supporting**. Used to encrypt data for remediations. |
| AWS Config | **Supporting**. Records all resources for use with AWS Security Hub. |

# Plan your deployment

This section describes the cost, network security, supported AWS Regions, quotas, and other considerations prior to deploying the solution.

## Cost

You are responsible for the cost of the AWS services used to run this solution.

As of this revision, the estimated monthly costs are:

- Small deployment (10 accounts, 1 region - US East/N. Virginia): Approximately $21.17 for 300 remediations/month
- Medium deployment (100 accounts, 1 region - US East/N. Virginia): Approximately $134.86 for 3,000 remediations/month
- Large deployment (1,000 accounts, 10 regions): Approximately $10,271.70 for 30,000 remediations/month

**Important**

Prices are subject to change. For full details, refer to the pricing page for each AWS service used in this solution.

**Note**

Many AWS Services include a Free Tier - a baseline amount of the service that customers can use at no charge. Actual costs may be more or less than the pricing examples provided.

We recommend creating a budget through AWS Cost Explorer to help manage costs. Prices are subject to change. For full details, see the pricing webpage for each AWS service used in this solution.

### Sample cost table

The total cost to run this solution depends on the following factors:

- The number of AWS Security Hub member accounts
- The number of active automatically-invoked remediations
- The frequency of remediation

This solution uses the following AWS components, which incur a cost based on your configuration. Pricing examples are provided for small, medium, and large organizations.

| Service | Free Tier | Pricing [USD] |
|---|---|---|
| AWS Systems Manager Automation - Step Count | 100,000 steps per account per month | Beyond the free tier, each basic step is charged at $0.002 per step. For multi-account automations, all steps including those run in any child accounts are counted only in the originating account. |
| AWS Systems Manager Automation - Step Duration | 5,000 seconds per month | Beyond the free tier, each aws:executeScript action step is charged at $0.00003 for every second after a free tier of 5,000 seconds per month. |
| AWS Systems Manager Automation - Storage | No free tier | $0.046 per GB per month |
| AWS Systems Manager Automation - Data Transfer | No free tier | $0.900 per GB transferred (for cross-account or out-of-Region) |
| AWS Security Hub - Security Checks | No free tier | First 100,000 checks/account/Region/month costs $0.0010 per check |

| Service | Free Tier | Pricing [USD] |
| --- | --- | --- |
| | | Next 400,000 checks/account/Region/month costs $0.0008 per check<br><br>Over 500,000 checks/account/Region/month costs $0.0005 per check |
| AWS Security Hub - Finding Ingestion Events | First 10,000 events/account/Region/month is free. Finding ingestion events associated with Security Hub's security checks. | Over 10,000 events/account/Region/month costs $0.00003 per event |
| Amazon CloudWatch - Metrics | Basic Monitoring Metrics (at 5-minute frequency) 10 Detailed Monitoring Metrics (at 1-minute frequency) 1 Million API requests (not applicable to GetMetricData and GetMetricWidgetImage) | First 10,000 metrics costs $0.30 metric/month<br><br>Next 240,000 metrics costs $0.10 metric/month<br><br>Next 750,000 metrics costs $0.05 metric/month<br><br>Over 1,000,000 metrics costs $0.02 metric/month<br><br>API calls cost $0.01 per 1,000 requests |
| Amazon CloudWatch - Dashboard | 3 Dashboards for up to 50 metrics per month | $3.00 per dashboard per month |

| Service | Free Tier | Pricing [USD] |
| --- | --- | --- |
| Amazon CloudWatch - Alarms | 10 Alarm metrics (not applicable to high-resolution alarms) | Standard Resolution (60 sec) costs $0.10 per alarmmetric<br><br>High Resolution (10 sec) costs $0.30 per alarm metric<br><br>Standard Resolution Anomaly Detection costs $0.30 per alarm<br><br>High Resolution Anomaly Detection costs $0.90 per alarm<br><br>Composite costs $0.50 per alarm |
| Amazon CloudWatch - Logs Collection | 5GB Data (ingestion, archive storage, and data scanned by Logs Insights queries) | $0.50 per GB |
| Amazon CloudWatch - Logs Storage | 5GB Data (ingestion, archive storage, and data scanned by Logs Insights queries) | $0.005 per GB of data scanned |
| Amazon CloudWatch - Events | All events except custom events are included | $1.00 per million events for custom events $1.00 per million events for cross-acc ount events |
| AWS Lambda - Requests | 1M free requests per month | $0.20 per 1M requests |

| Service | Free Tier | Pricing [USD] |
| --- | --- | --- |
| AWS Lambda - Duration | 400,000 GB-seconds of compute time per month | $0.0000166667 for every GB-second. The price for Duration depends on the amount of memory you allocate to your function. You can allocate any amount of memory to your function between 128MB and 10,240MB, in 1MB increments. |
| AWS Step Functions - State Transitions | 4,000 free state transitions per month | $0.025 per 1,000 state transitions thereafter |
| Amazon EventBridge | All state change events published by AWS services are free | Custom events cost $1.00/million custom events published<br><br>Third-party (SaaS) events cost $1.00/million events published<br><br>Cross-account events cost $1.00/million cross-account events sent |
| Amazon SNS | First 1 million Amazon SNS requests per month are free | $0.50 per 1 million requests thereafter |
| Amazon SQS | First 1 million Amazon SQS requests per month are free | $0.40 per 1 million to 100 billion requests thereafter |
| Amazon DynamoDB | First 25GB of storage is free | $2.00 per 1 million consistent reads and writes thereafter |

| Service | Free Tier | Pricing [USD] |
| --- | --- | --- |
| AWS Key Management Service pricing | 20,000 requests/month | $1.00 per 1 KMS key. For KMS keys that you rotate automatically or on demand, the first and second rotation of the key adds $1/month (prorated hourly) in cost. |

## Pricing examples (monthly)

### Example 1: 300 remediations per month

- 10 accounts, 1 Region
- 30 remediations per account/Region/month
- Total cost $21.17 per month

| Service | Assumptions | Monthly charges [USD] |
| --- | --- | --- |
| AWS Systems Manager Automation | Steps: ~4 steps * 300 remediations * $0.002 = $2.40<br><br>Duration: 10s * 300 remediations * $0.00003 = $0.09 | $2.49 |
| AWS Security Hub | No billable services utilized | $0 |
| Amazon CloudWatch Logs | 300 remediations * $0.000002 = $0.0006<br><br>$0.0006 * 0.03 = $0.000018 | < $0.01 |
| AWS Lambda - Requests | 300 remediations * 6 requests = 1,800 requests | $0.20 |

| Service | Assumptions | Monthly charges [USD] |
|---|---|---|
| | $0.20 * 1,000,000 requests = $0.20 | |
| AWS Lambda - Duration | 256M: 1.875 GB sec * 300 remediations * $0.0000167 = $0.009375 | < $0.01 |
| AWS Step Functions | 17 state transitions * 300 remediations = 5,100<br><br>$0.025 * (5,100/1,000) state transitions = $0.15 | $0.15 |
| Amazon EventBridge rules | No charge for rules | $0 |
| AWS Key Management Service | 1 key * 10 accounts * 1 Region * $1 = $10 | $10.00 |
| Amazon DynamoDB | $2.00 * 1,000,000 read and writes = $2.00 | $2.00 |
| Amazon SQS | $0.40 * 1,000,000 requests = $0.40 | $0.40 |
| Amazon SNS | $0.50 * 1,000,000 notificat ions = $0.50 | $0.50 |
| Amazon CloudWatch - Metrics | $0.30 * 7 custom metrics = $2.10<br><br>$0.01 * (300 * 3 / 1,000) put metrics API calls = $0.01 | $2.11 |
| Amazon CloudWatch - Dashboards | $3.00 * 1 dashboard = $3.00 | $3.00 |
| Amazon CloudWatch - Alarms | $0.10 * 3 alarms = $0.30 | $0.30 |

| Service | Assumptions | Monthly charges [USD] |
|---|---|---|
| **Total** | | **$21.17** |

## Example 2: 3,000 remediations per month

- 100 accounts, 1 Region
- 30 remediations per account/Region/month
- Total cost $134.86 per month

| Service | Assumptions | Monthly charges [USD] |
|---|---|---|
| AWS Systems Manager Automation | Steps: ~4 steps * 3,000 remediations * $0.002 = $24.00<br><br>Duration: 10s * 3,000 remediations * $0.00003 = $0.90 | $24.90 |
| AWS Security Hub | No billable services utilized | $0 |
| Amazon CloudWatch Logs | 3,000 remediations * $0.000002 = $0.006<br><br>$0.006 * 0.03 = $0.00018 | < $0.01 |
| AWS Lambda - Requests | 3,000 remediations * 6 requests = 18,000 requests<br><br>$0.20 * 1,000,000 requests = $0.20 | $0.20 |
| AWS Lambda - Duration | 256M: 1.875 GB sec * 3,000 remediations * $0.000167 = $0.09375 | $0.09 |

| Service | Assumptions | Monthly charges [USD] |
| --- | --- | --- |
| AWS Step Functions | 17 state transitions * 3,000 remediations = 51,000<br><br>$0.025 * (51,000/1,000) state transitions = $1.275 | $1.28 |
| Amazon EventBridge rules | No charge for rules | $0 |
| AWS Key Management Service | 1 key * 100 accounts * 1 Region * $1 = $100 | $100 |
| Amazon DynamoDB | $2.00 * 1,000,000 read and writes = $2.00 | $2.00 |
| Amazon SQS | $0.40 * 1,000,000 requests = $0.40 | $0.40 |
| Amazon SNS | $0.50 * 1,000,000 notifications = $0.50 | $0.50 |
| Amazon CloudWatch - Metrics | $0.30 * 7 custom metrics = $2.10<br><br>$0.01 * (3000 * 3 / 1,000) put metrics API calls = $0.09 | $2.19 |
| Amazon CloudWatch - Dashboards | $3.00 * 1 dashboard = $3.00 | $3.00 |
| Amazon CloudWatch - Alarms | $0.10 * 3 alarms = $0.30 | $0.30 |
| **Total** | | **$134.86** |

## Example 3: 30,000 remediations per months

- 1,000 accounts, 10 Regions
- 30 remediations per account/Region/month

- Total cost $1,271.70 per month

| Service | Assumptions | Monthly charges [USD] |
|---|---|---|
| AWS Systems Manager Automation | Steps: ~4 steps * 30,000 remediations * $0.002 = $240.00<br><br>Duration: 10s * 30,000 remediations * $0.00003 = $9.00 | $249.00 |
| AWS Security Hub | No billable services utilized | $0 |
| Amazon CloudWatch Logs | 30,000 remediations * $0.000002 = $0.06<br><br>$0.06 * 0.03 = $0.0018 | < $0.01 |
| AWS Lambda - Requests | 30,000 remediations * 6 requests = 180,000 requests<br><br>$0.20 * 1,000,000 requests = $0.20 | $0.20 |
| AWS Lambda - Duration | 256M: 1.875 GB sec * 30,000 remediations * $0.000167 = $0.9375 | $0.94 |
| AWS Step Functions | 17 state transitions * 30,000 remediations = 510,000<br><br>$0.025 * (510,000/1,000) state transitions = $12.75 | $12.75 |
| Amazon EventBridge rules | No charge for rules | $0 |
| AWS Key Management Service | (1 key) $1 * 1,000 accounts * 10 Region = $10,000 | $10,000 |

| Service | Assumptions | Monthly charges [USD] |
| --- | --- | --- |
| Amazon DynamoDB | $0.000002 * 1,000,000 read and writes = $2.00 | $2.00 |
| Amazon SQS | $0.000004 * 1,000,000 requests = $0.40 | $0.40 |
| Amazon SNS | $0.000005 * 1,000,000 notifications = $0.50 | $0.50 |
| Amazon CloudWatch - Metrics | $0.30 * 6 custom metrics = $1.80<br><br>$0.01 * (30,000 * 3 / 1,000) put metrics API calls = $0.90 | $2.70 |
| Amazon CloudWatch - Dashboards | $3.00 * 1 dashboard = $3.00 | $3.00 |
| Amazon CloudWatch - Alarms | $0.10 * 2 alarms = $0.20 | $0.20 |
| **Total** | | **$10,271.70** |

**Important**

**KMS Key Rotation Costs** AWS Key Management Service (KMS) automatically rotates customer managed keys once per year when rotation is enabled. Each rotation incurs a cost of $1.00 per key per year. For example, with 1000 accounts in a single region, this results in an additional $1000/year (1 rotation × 1000 keys × $1.00).

## Additional cost for optional features

This section identifies additional costs associated with optional features for this solution.

## Enhanced CloudWatch metrics

If you select `yes` for the **EnableEnhancedCloudWatchMetrics** parameter when deploying the admin stack, the solution creates two custom metrics and one alarm for each control ID. The cost depends on the number of control IDs that you are remediating. In the following table, we assume that you are remediating all 96 different control IDs per month, to determine the upper bound of costs.

| Service | Assumptions 96 control IDs * 2 = 192 custom metrics | Monthly charges [USD] |
|---|---|---|
| Amazon CloudWatch - Metrics | $0.30 * 192 custom metrics = $57.60 | $57.60 |
| Amazon CloudWatch - Alarms | $0.10 * 96 alarms = $9.60 | $9.60 |
| **Total** | | **$67.20** |

## CloudTrail Action Log

In each member account that you enable the Action Log feature for, the solutions creates a CloudTrail trail to log all write management events. A Lambda function filters out events not related to the solution. This means that the cost is related to the total number of management events in your account, since events not related to the solution are still captured by the trail and processed by the Lambda function.

For the following table, we assume 150,000 management events per month in the account. The actual cost depends on the actual management event activity in your account.

| Service | Assumptions | Monthly charges [USD] |
|---|---|---|
| AWS CloudTrail | 150,000 * $2.00/100,000 = $3.00 | $3.00 |
| Lambda | 150,000 * 0.2 * 0.125 = 3,750 GB-seconds | $0.0925 |

| Service | Assumptions | Monthly charges [USD] |
| --- | --- | --- |
| | 3,750 * $0.0000166667 = $0.0625 compute time cost<br><br>0.15 * $0.20 = $0.03 request cost<br><br>$0.0625 + $0.03 = $0.0952 total Lambda cost | |
| **Total** | | **$3.09 per member account** |

# Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information about AWS security, visit the AWS Cloud Security.

# IAM roles

AWS Identity and Access Management (IAM) roles allow customers to assign granular access policies and permissions to services and users in the AWS Cloud. This solution creates IAM roles that grant the solution's automated functions access to perform remediation actions within a narrow scope set of permissions specific to each remediation.

The admin account's Step Function is assigned to the SO0111-ASR-Orchestrator-Admin role. Only this role is allowed to assume the SO0111-Orchestrator-Member in each member account. The member role is allowed by each remediation role to pass it to the AWS Systems Manager service to run specific remediation runbooks. Remediation role names begin with SO0111, followed by a description matching the name of the remediation runbook. For example, SO0111-RemoveVPCDefaultSecurityGroupRules is the role for the ASR-RemoveVPCDefaultSecurityGroupRules remediation runbook.

# Supported AWS Regions

| Region name | Region code |
|---|---|
| US East (Ohio) | us-east-2 |
| US East (N. Virginia) | us-east-1 |
| US West (Northern California) | us-west-1 |
| US West (Oregon) | us-west-2 |
| Africa (Cape Town) | af-south-1 |
| Asia Pacific (Hong Kong) | ap-east-1 |
| Asia Pacific (Hyderabad) | ap-south-2 |
| Asia Pacific (Jakarta) | ap-southeast-3 |
| Asia Pacific (Melbourne) | ap-southeast-4 |
| Asia Pacific (Mumbai) | ap-south-1 |
| Asia Pacific (Osaka) | ap-northeast-3 |
| Asia Pacific (Seoul) | ap-northeast-2 |
| Asia Pacific (Singapore) | ap-southeast-1 |
| Asia Pacific (Sydney) | ap-southeast-2 |
| Asia Pacific (Tokyo) | ap-northeast-1 |
| Canada (Central) | ca-central-1 |
| Europe (Frankfurt) | eu-central-1 |
| Europe (Ireland) | eu-west-1 |
| Europe (London) | eu-west-2 |

| Region name | Region code |
|---|---|
| Europe (Milan) | eu-south-1 |
| Europe (Paris) | eu-west-3 |
| Europe (Spain) | eu-south-2 |
| Europe (Stockholm) | eu-north-1 |
| Europe (Zurich) | eu-central-2 |
| Middle East (Bahrain) | me-south-1 |
| Middle East (UAE) | me-central-1 |
| South America (Sao Paulo) | sa-east-1 |
| AWS GovCloud (US-East) | us-gov-east-1 |
| AWS GovCloud (US-West) | us-gov-west-1 |
| China (Beijing) | cn-north-1 |
| China (Ningxia) | cn-northwest-1 |
| Israel (Tel Aviv) | il-central-1 |
| Canada West (Calgary) | ca-west-1 |
| Mexico (Mexico City) | mx-central-1 |
| Asia Pacific (Thailand) | ap-southeast-7 |

**Note**

Any new AWS regions not listed may be supported via local deployment but not one-click deployment.

# Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

## Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the services implemented in this solution. For more information, refer to AWS service quotas.

Use the following links to go to the page for that service. To view the Service Quotas for all AWS services in the documentation without switching pages, view the information in the Service endpoints and quotas page in the PDF instead.

## AWS CloudFormation quotas

Your AWS account has AWS CloudFormation quotas that you should be aware of when launching the stack in this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, see AWS CloudFormation quotas in the *AWS CloudFormation User Guide*.

## AWS CloudWatch quotas

Your AWS account has AWS CloudWatch quotas tied to CloudWatch Resource Policies which only allows 10 resource policies per region per account and this cannot be requested for a quota increase, see AWS CloudWatch Logs Quotas in the *AWS CloudWatch User Guide*. Before your deployment please check your current usage to ensure you won't cross this threshold when deploying the solution.

## Amazon EventBridge rules quotas

Your AWS account has Amazon EventBridge rules quotas that you should be aware of when selecting the playbooks to deploy with the solution. Each playbook will create an EventBridge Rule for each control it can remediate. When deploying multiple playbooks, it is possible to reach the quota for Rules. For more information, see Amazon EventBridge quotas in the *Amazon EventBridge User Guide*.

# AWS Security Hub deployment

AWS Security Hub deployment and configuration is a prerequisite for this solution. For more information about setting up AWS Security Hub, refer to Setting up AWS Security Hub in the *AWS Security Hub User Guide.*

At minimum, you must have a working Security Hub configured in your primary account. You can deploy this solution in the same account (and AWS Region) as the Security Hub primary account. In each Security Hub primary and secondary account, you must also deploy the member template that allows AssumeRole permissions to the solution's AWS Step Functions to run remediation runbooks in the account.

# Stack vs StackSets deployment

A *stack set* lets you create stacks in AWS accounts across AWS Regions by using a single AWS CloudFormation template. Starting with version 1.4, this solution supports stack set deployment by splitting resources based on where and how they are deployed. Multi-account customers, particularly those using AWS Organizations, can benefit from using stack sets for deployment across many accounts. It reduces the effort needed to install and maintain the solution. For more information about StackSets, refer to Using AWS CloudFormation StackSets.

# Deploy the solution

**Important**

If the consolidated control findings feature is turned on in Security Hub (**this is default in new deployments**), only enable the Security Control (CS) playbook **when deploying this solution**. If the feature is not turned on, **only** enable the playbooks for the security standards that are enabled in Security Hub. Enabling additional playbooks can result in reaching the quota for EventBridge Rules.

This solution uses AWS CloudFormation templates and stacks to automate its deployment. The CloudFormation templates specify the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources that are described in the templates.

In order for the solution to function, three templates must be deployed. First, decide where to deploy the templates, then decide how to deploy them.

This overview will describe the templates and how to decide where and how to deploy them. The next sections will have more detailed instructions for deploying each stack as a Stack or StackSet.

## Deciding where to deploy each stack

The three templates will be referred to by the following names and contain the following resources:

- Admin stack: orchestrator step function, event rules and Security Hub custom action.
- Member stack: remediation SSM Automation documents.
- Member roles stack: IAM roles for remediations.

The Admin stack must be deployed once, in a single account and a single Region. It must be deployed into the account and Region that you have configured as the aggregation destination for Security Hub findings for your organization. If you wish to use the Action Log feature to monitor management events, you must deploy the Admin stack in your organization's management account or a delegated administrator account.

The solution operates on Security Hub findings, so it will not be able to operate on findings from a particular account and Region if that account or Region has not been configured to aggregate findings in the Security Hub administrator account and Region.

For example, an organization has accounts operating in Regions `us-east-1` and `us-west-2`, with account `111111111111` as the Security Hub delegated administrator in Region `us-east-1`. Accounts `222222222222` and `333333333333` must be Security Hub member accounts for the delegated administrator account `111111111111`. All three accounts must be configured to aggregate findings from `us-west-2` to `us-east-1`. The Admin stack must be deployed to account `111111111111` in `us-east-1`.

For more details on finding aggregation, consult the documentation for Security Hub delegated administrator accounts and cross-Region aggregation.

The Admin stack must complete deployment first before deploying the member stacks so that a trust relationship can be created from the member accounts to the hub account.

The member stack must be deployed into every account and Region in which you wish to remediate findings. This can include the Security Hub delegated administrator account in which you previously deployed the ASR Admin stack.The automation documents must execute in the member accounts in order to use the free tier for SSM Automation.

Using the previous example, if you want to remediate findings from all accounts and Regions, the member stack must be deployed to all three accounts (`111111111111`, `222222222222`, and `333333333333`) and both Regions (`us-east-1` and `us-west-2`).

The member roles stack must be deployed to every account, but it contains global resources (IAM roles) that can only be deployed once per account. It does not matter in which Region you deploy the member roles stack, so for simplicity we suggest deploying to the same Region in which the Admin stack is deployed.

Using the previous example, we suggest deploying the member roles stack to all three accounts (`111111111111`, `222222222222`, and `333333333333`) in `us-east-1`.

## Deciding how to deploy each stack

The options for deploying a stack are

- CloudFormation StackSet (self-managed permissions)
- CloudFormation StackSet (service-managed permissions)

- CloudFormation Stack

StackSets with service-managed permissions are the most convenient because they do not require deploying your own roles and can automatically deploy to new accounts in the organization. Unfortunately, this method does not support nested stacks, which we use in both the Admin stack and the member stack. The only stack that can be deployed this way is the member roles stack.

Be aware that when deploying to the entire organization, the organization management account is not included, so if you want to remediate findings in the organization management account, you must deploy to this account separately.

The member stack must be deployed to every account and Region but cannot be deployed using StackSets with service-managed permissions because it contains nested stacks. So we suggest deploying this stack with StackSets with self-managed permissions.

The Admin stack is only deployed once, so it can be deployed as a plain CloudFormation stack or as a StackSet with self-managed permissions in a single account and Region.

## Consolidated control findings

The accounts in your organization can be configured with the consolidated control findings feature of Security Hub turned on or off. See Consolidated control findings in the *AWS Security Hub User Guide*.

> **Important**
>
> If enabled, you must use v2.0.0 of the solution or later. In addition, you must deploy both the Admin and Member nested stacks for the "SC" or "security control" standards. This deploys the automation documents and EventBridge rules for use with the consolidated control IDs generated when this feature is turned on. There is no need to deploy the Admin or Member nested stacks for specific standards (e.g. AWS FSBP) when using this feature.

# AWS CloudFormation templates

View template

**automated-security-response-admin.template** - Use this template to launch the Automated

Security Response on AWS solution. The template installs the core components of the solution, a nested stack for the AWS Step Functions logs, and one nested stack for each security standard you choose to activate.

Services used include Amazon Simple Notification Service, AWS Key Management Service, AWS Identity and Access Management, AWS Lambda, AWS Step Functions, Amazon CloudWatch Logs, Amazon S3, and AWS Systems Manager.

## Admin account support

The following templates are installed in the AWS Security Hub admin account to turn on the security standards that you want to support. You can choose which of the following templates to install when installing the `automated-security-response-admin.template`.

**automated-security-response-orchestrator-log.template** - Creates a CloudWatch logs group for the Orchestrator Step Function.

**AFSBPStack.template** - AWS Foundational Security Best Practices v1.0.0 rules.

**CIS120Stack.template** - CIS Amazon Web Services Foundations benchmarks, v1.2.0 rules.

**CIS140Stack.template** - CIS Amazon Web Services Foundations benchmarks, v1.4.0 rules.

**CIS300Stack.template** - CIS Amazon Web Services Foundations benchmarks, v3.0.0 rules.

**PCI321Stack.template** - PCI-DSS v3.2.1 rules.

**NISTStack.template** - National Institute of Standards and Technology (NIST), v5.0.0 rules.

**SCStack.template** - Security Controls v2.0.0 rules.

## Member roles


**automated-security-response-member-roles.template** - Defines the remediation roles needed in each AWS Security Hub member account.

## Member accounts

View template

**automated-security-response-member.template** - Use this template after you set up the core solution to install AWS Systems Manager automation runbooks and permissions in each of your AWS Security Hub member accounts (including the admin account). This template allows you to choose which security standard playbooks to install.

The `automated-security-response-member.template` installs the following templates based on your selections:

**automated-security-response-remediation-runbooks.template** - Common remediation code used by one or more of the security standards.

**AFSBPMemberStack.template** - AWS Foundational Security Best Practices v1.0.0 settings, permissions, and remediation runbooks.

**CIS120MemberStack.template** - CIS Amazon Web Services Foundations benchmarks, version 1.2.0 settings, permissions, and remediation runbooks.

**CIS140MemberStack.template** - CIS Amazon Web Services Foundations benchmarks, version 1.4.0 settings, permissions, and remediation runbooks.

**CIS300MemberStack.template** - CIS Amazon Web Services Foundations benchmarks, version 3.0.0 settings, permissions, and remediation runbooks.

**PCI321MemberStack.template** - PCI-DSS v3.2.1 settings, permissions, and remediation runbooks.

**NISTMemberStack.template** - National Institute of Standards and Technology (NIST), v5.0.0 settings, permissions, and remediation runbooks.

**SCMemberStack.template** - Security Control settings, permissions, and remediation runbooks.

**automated-security-response-member-cloudtrail.template** - Used in the Action Log feature to track and audit and service activity.

## Ticket system integration

Use one of the following templates to integrate with your ticketing system.

View template

**JiraBlueprintStack.template** - Deploy if you use Jira as your ticketing system.

View template

**ServiceNowBlueprintStack.template** - Deploy if you use ServiceNow as your ticketing system.

If you want to integrate a different external ticketing system, you can use either of these stacks as blueprint to understand how to implement your own custom integration.

# Automated deployment - StackSets

> **Note**
>
> We recommend deploying with StackSets. However, for single account deployments or for testing or evaluation purposes, consider the stacks deployment option.

Before you launch the solution, review the architecture, solution components, security, and design considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your AWS Organizations.

**Time to deploy:** Approximately 30 minutes per account, depending upon StackSet parameters.

## Prerequisites

AWS Organizations helps you centrally manage and govern your multi-account AWS environment and resources. StackSets work best with AWS Organizations.

If you have previously deployed v1.3.x or earlier of this solution, you must uninstall the existing solution. For more information, refer to Update the solution.

Before you deploy this solution, review your AWS Security Hub deployment:

- There must be a delegated Security Hub admin account in your AWS Organization.
- Security Hub should be configured to aggregate findings across Regions. For more information, refer to Aggregating findings across Regions in the AWS Security Hub User Guide.

- You should activate Security Hub for your organization in each Region where you have AWS usage.

This procedure assumes that you have multiple accounts using AWS Organizations, and have delegated an AWS Organizations admin account and an AWS Security Hub admin account.

## Deployment overview

> **Note**
>
> StackSets deployment for this solution uses a combination of service-managed and self-managed StackSets. Self-Managed StackSets must be used currently as they use nested StackSets, which are not yet supported with service-managed StackSets.

Deploy the StackSets from a delegated administrator account in your AWS Organizations.

### Planning

Use the following form to help with StackSets deployment. Prepare your data, then copy and paste the values during deployment.

```
AWS Organizations admin account ID: _________________
Security Hub admin account ID: _________________
CloudTrail Logs Group: ____________________________
Member account IDs (comma-separated list):
____________________,
____________________,
____________________,
____________________,
____________________
AWS Organizations OUs (comma-separated list):
____________________,
____________________,
____________________,
____________________,
____________________
```

(Optional) Step 0: Deploy the ticketing integration stack

- If you intend to use the ticketing feature, deploy the ticketing integration stack into your Security Hub admin account first.
- Copy the Lambda function name from this stack and provide it as input to the admin stack (see Step 1).

### Step 1: Launch the admin stack in the delegated Security Hub admin account

- Using a self-managed StackSet, launch the `automated-security-response-admin.template` AWS CloudFormation template into your AWS Security Hub admin account in the same Region as your Security Hub admin. This template uses nested stacks.
- Choose which Security Standards to install. By default, only SC is selected (Recommended).
- Choose an existing Orchestrator log group to use. Select `Yes` if `SO0111-ASR- Orchestrator` already exists from a previous installation.

For more information on self-managed StackSets, refer to Grant self-managed permissions in the *AWS CloudFormation User Guide*.

### Step 2: Install the remediation roles into each AWS Security Hub member account

Wait for Step 1 to complete deployment, because the template in Step 2 references IAM roles created by Step 1.

- Using a service-managed StackSet, launch the `automated-security-response-member-roles.template` AWS CloudFormation template into a single Region in each account in your AWS Organizations.
- Choose to install this template automatically when a new account joins the organization.
- Enter the account ID of your AWS Security Hub admin account.

### Step 3: Launch the member stack into each AWS Security Hub member account and Region

- Using self-managed StackSets, launch the `automated-security-response-member.template` AWS CloudFormation template into all Regions where you have AWS resources in every account in your AWS Organization managed by the same Security Hub admin.

> **Note**
>
> Until service-managed StackSets support nested stacks, you must do this step for any new accounts that join the organization.

- Choose which Security Standard playbooks to install.
- Provide the name of a CloudTrail logs group (used by some remediations).
- Enter the account ID of your AWS Security Hub admin account.

## (Optional) Step 0: Launch a ticket system integration stack

1. If you intend to use the ticketing feature, launch the respective integration stack first.
2. Choose the provided integration stacks for Jira or ServiceNow, or use them as a blueprint to implement your own custom integration.

   **To deploy the Jira stack**:

   a. Enter a name for your stack.

   b. Provide the URI to your Jira instance.

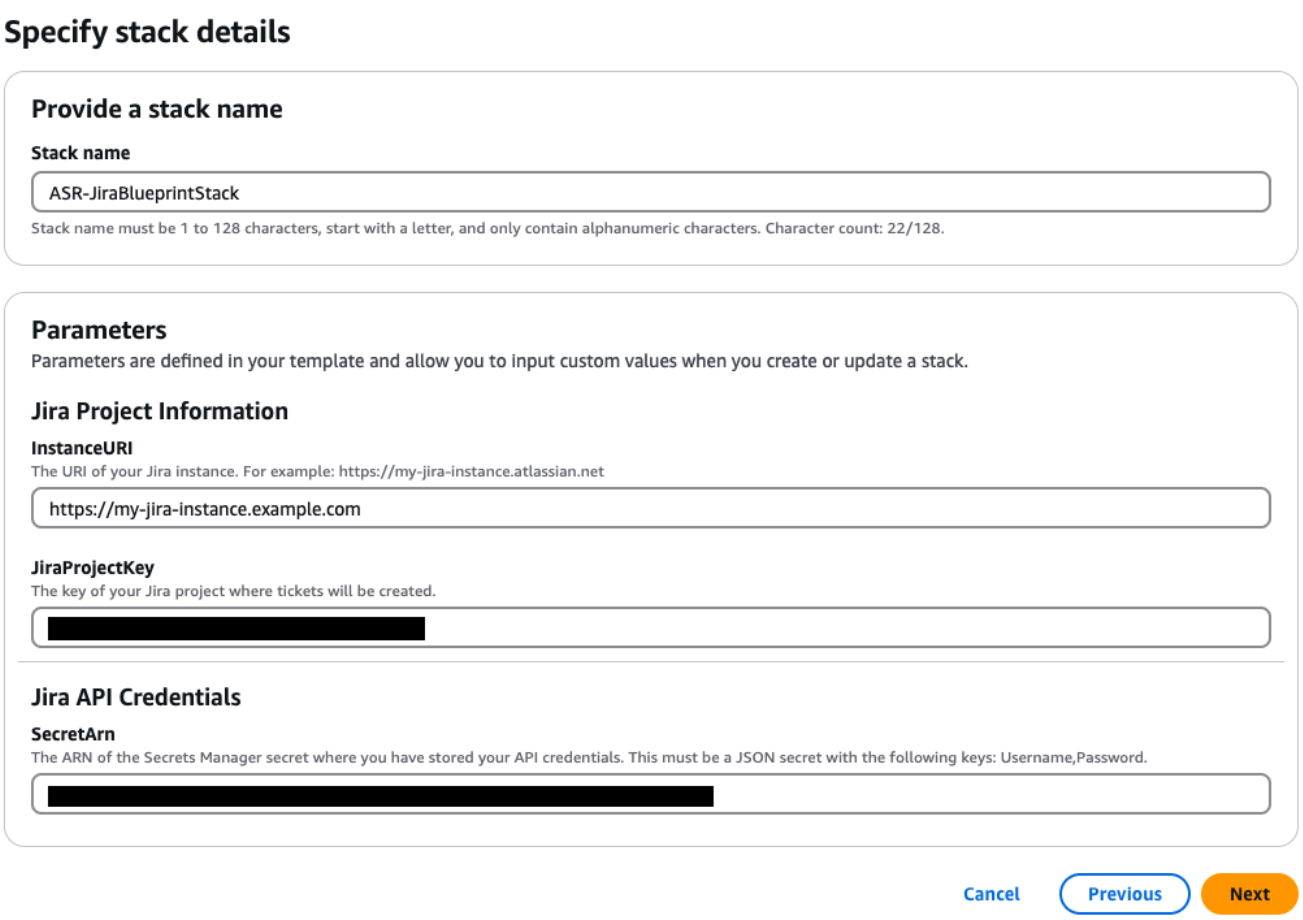
   c. Provide the project key for the Jira project that you want to send tickets to.

   d. Create a new key-value secret in Secrets Manager that holds your Jira `Username` and `Password`.

   > **Note**
   >
   > You can choose to use a Jira API key in place of your password by providing your username as `Username` and your API key as the `Password`.

   e. Add the ARN of this secret as input to the stack.

   **Provide a stack name Jira project information, and Jira API credentials.**




**To deploy the ServiceNow stack**:

f. Enter a name for your stack.

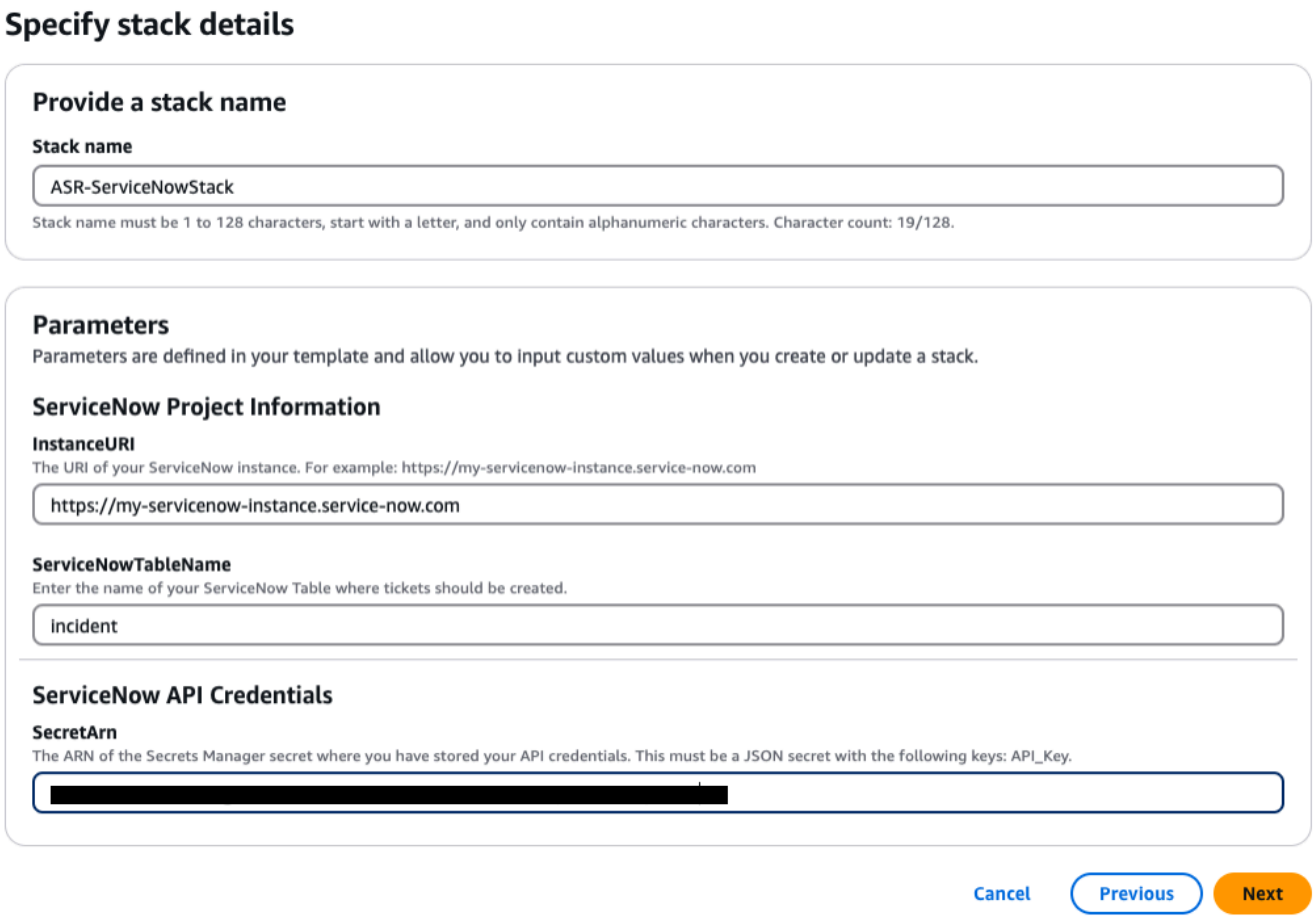
g. Provide the URI of your ServiceNow instance.

h. Provide your ServiceNow table name.

i. Create an API key in ServiceNow with permission to modify the table you intend to write to.

j. Create a secret in Secrets Manager with the key `API_Key` and provide the secret ARN as input to the stack.

**Provide a stack name ServiceNow project information, and ServiceNow API credentials.**

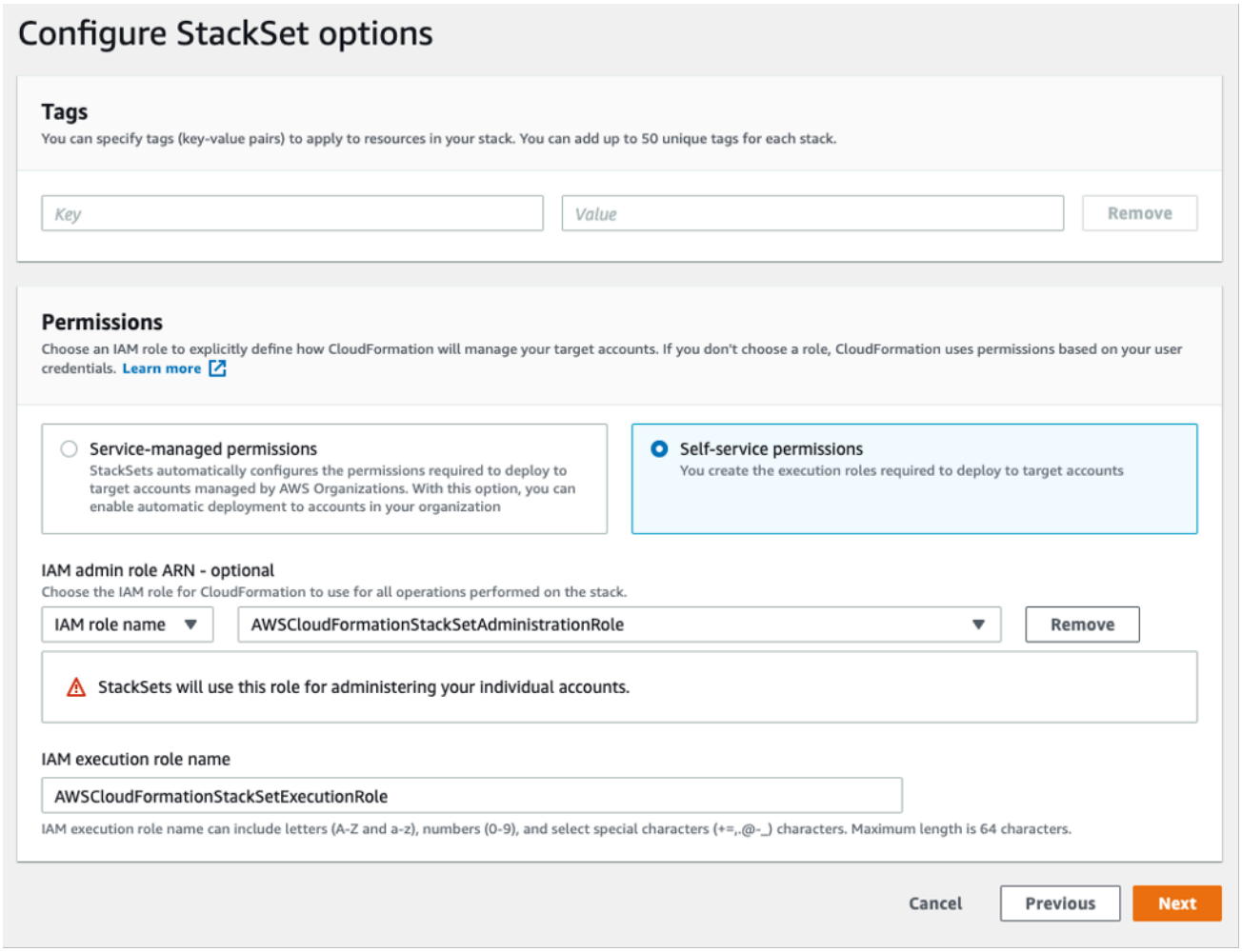



**To create a custom integration stack**: Include a Lambda function that the solution orchestrator Step Functions can call for each remediation. The Lambda function should take the input provided by Step Functions, construct a payload according to the requirements of your ticketing system, and make a request to your system to create the ticket.

## Step 1: Launch the admin stack in the delegated Security Hub admin account

1. Launch the admin stack, `automated-security-response-admin.template`, with your Security Hub admin account. Typically, one per organization in a single Region. Because this stack uses nested stacks, you must deploy this template as a self-managed StackSet.

   **Configure StackSet options**




2. For the **Account numbers** parameter, enter the account ID of the AWS Security Hub admin account.
3. For the **Specify regions** parameter, select only the Region where Security Hub admin is turned on. Wait for this step to complete before going on to Step 2.

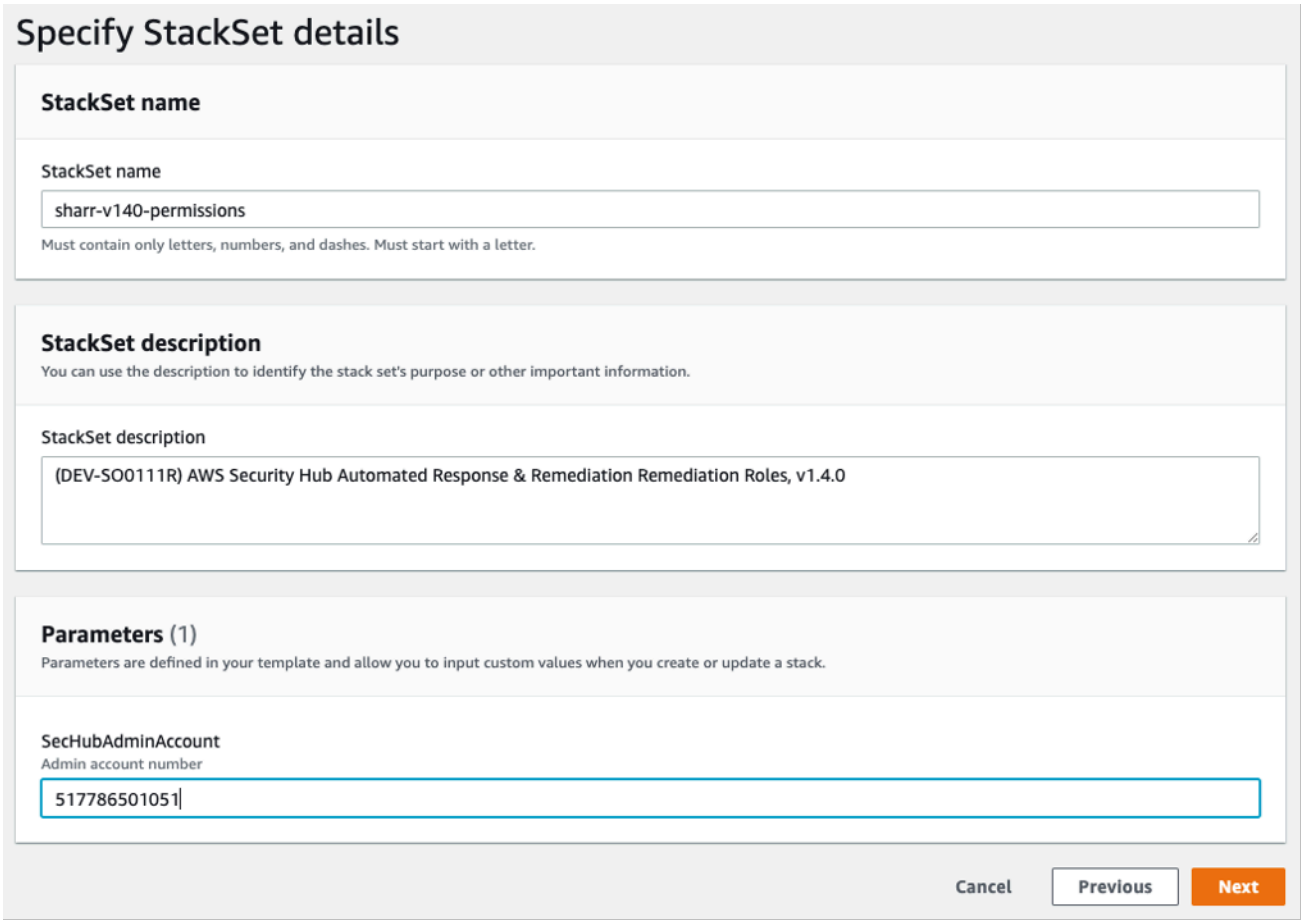
## Step 2: Install the remediation roles into each AWS Security Hub member account

Use a service-managed StackSets to deploy the member roles template, `automated-security-response-member-roles.template`. This StackSet must be deployed in one Region per member account. It defines the global roles that allow cross-account API calls from the ASR Orchestrator step function.

1. Deploy to the entire organization (typical) or to organizational units, as per your organizations policies.

2. Turn on automatic deployment so new accounts in the AWS Organizations receive these permissions.
3. For the **Specify regions** parameter, select a single Region. IAM roles are global. You can continue to Step 3 while this StackSet deploys.

**Specify StackSet details**




## Step 3: Launch the member stack into each AWS Security Hub member account and Region

Because the member stack uses nested stacks, you must deploy as a self-managed StackSet. This does not support automatic deployment to new accounts in the AWS Organization.

### Parameters

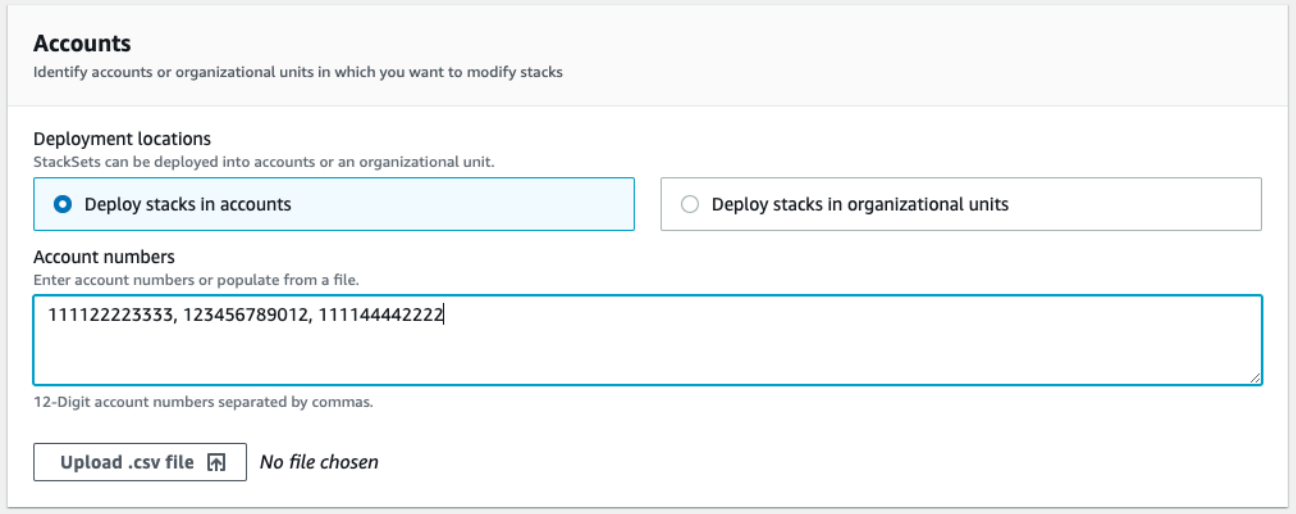
**LogGroup Configuration**: Choose the log group that receives CloudTrail logs. If none exists, or if the log group is different for each account, choose a convenient value. Account administrators

must update the Systems Manager - Parameter Store /Solutions/SO0111/Metrics_LogGroupName parameter after creating a CloudWatch Logs Group for CloudTrail logs. This is required for remediations that create metrics alarms on API calls.

**Standards**: Choose the standards to load in the member account. This only installs the AWS Systems Manager runbooks - it does not enable the Security Standard.

**SecHubAdminAccount**: Enter the account ID of the AWS Security Hub Admin account where you installed the solution's admin template.

**Accounts**




**Deployment locations**: You may specify a list of account numbers or organizational units.

**Specify regions**: Select all of the Regions where you want to remediate findings. You can adjust Deployment options as appropriate for the number of accounts and Regions. Region Concurrency can be parallel.

# Automated deployment - Stacks

> **Note**
>
> For multi-account customers, we strongly recommend deployment with StackSets.

Before you launch the solution, review the architecture, solution components, security, and design considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

**Time to deploy:** Approximately 30 minutes

## Prerequisites

Before you deploy this solution, ensure that AWS Security Hub is in the same AWS Region as your primary and secondary accounts. If you have previously deployed this solution, you must uninstall the existing solution. For more information, refer to Update the solution.

## Deployment overview

Use the following steps to deploy this solution on AWS.

### (Optional) Step 0: Launch a ticket system integration stack

- If you intend to use the ticketing feature, deploy the ticketing integration stack into your Security Hub admin account first.
- Copy the Lambda function name from this stack and provide it as input to the admin stack (see Step 1).

### Step 1: Launch the admin stack

- Launch the `automated-security-response-admin.template` AWS CloudFormation template into your AWS Security Hub admin account.
- Choose which security standards to install.
- Choose an existing Orchestrator log group to use (select `Yes` if `SO0111-ASR-Orchestrator` already exists from a previous installation).

### Step 2: Install the remediation roles into each AWS Security Hub member account

- Launch the `automated-security-response-member-roles.template` AWS CloudFormation template into one Region per member account.
- Enter the 12-digit account IG for the AWS Security Hub admin account.

### Step 3: Launch the member stack

- Specify the name of the CloudWatch Logs group to use with CIS 3.1-3.14 remediations. It must be the name of a CloudWatch Logs log group that receives CloudTrail logs.

- Choose whether to install the remediation roles. Install these roles only once per account.
- Select which playbooks to install.
- Enter the account ID of the AWS Security Hub admin account.

Step 4: (Optional) Adjust the available remediations

- Remove any remediations on a per-member account basis. This step is optional.

## (Optional) Step 0: Launch a ticket system integration stack

1. If you intend to use the ticketing feature, launch the respective integration stack first.
2. Choose the provided integration stacks for Jira or ServiceNow, or use them as a blueprint to implement your own custom integration.

   **To deploy the Jira stack**:

   a. Enter a name for your stack.

   b. Provide the URI to your Jira instance.

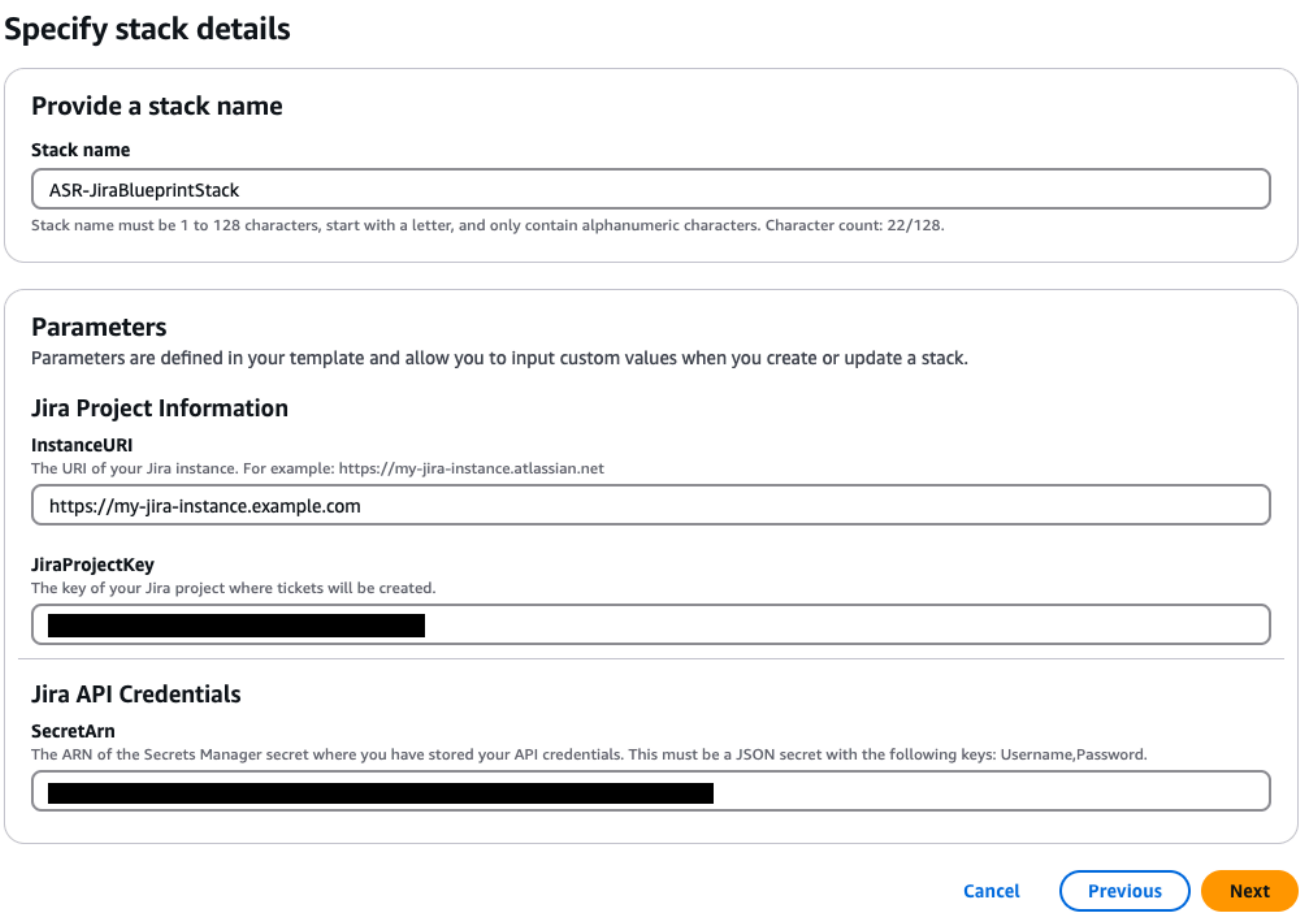
   c. Provide the project key for the Jira project that you want to send tickets to.

   d. Create a new key-value secret in Secrets Manager that holds your Jira `Username` and `Password`.

   > **Note**
   >
   > You can choose to use a Jira API key in place of your password by providing your username as `Username` and your API key as the `Password`.

   e. Add the ARN of this secret as input to the stack.

   **"Provide a stack name Jira project information, and Jira API credentials.**




**To deploy the ServiceNow stack**:

f. Enter a name for your stack.

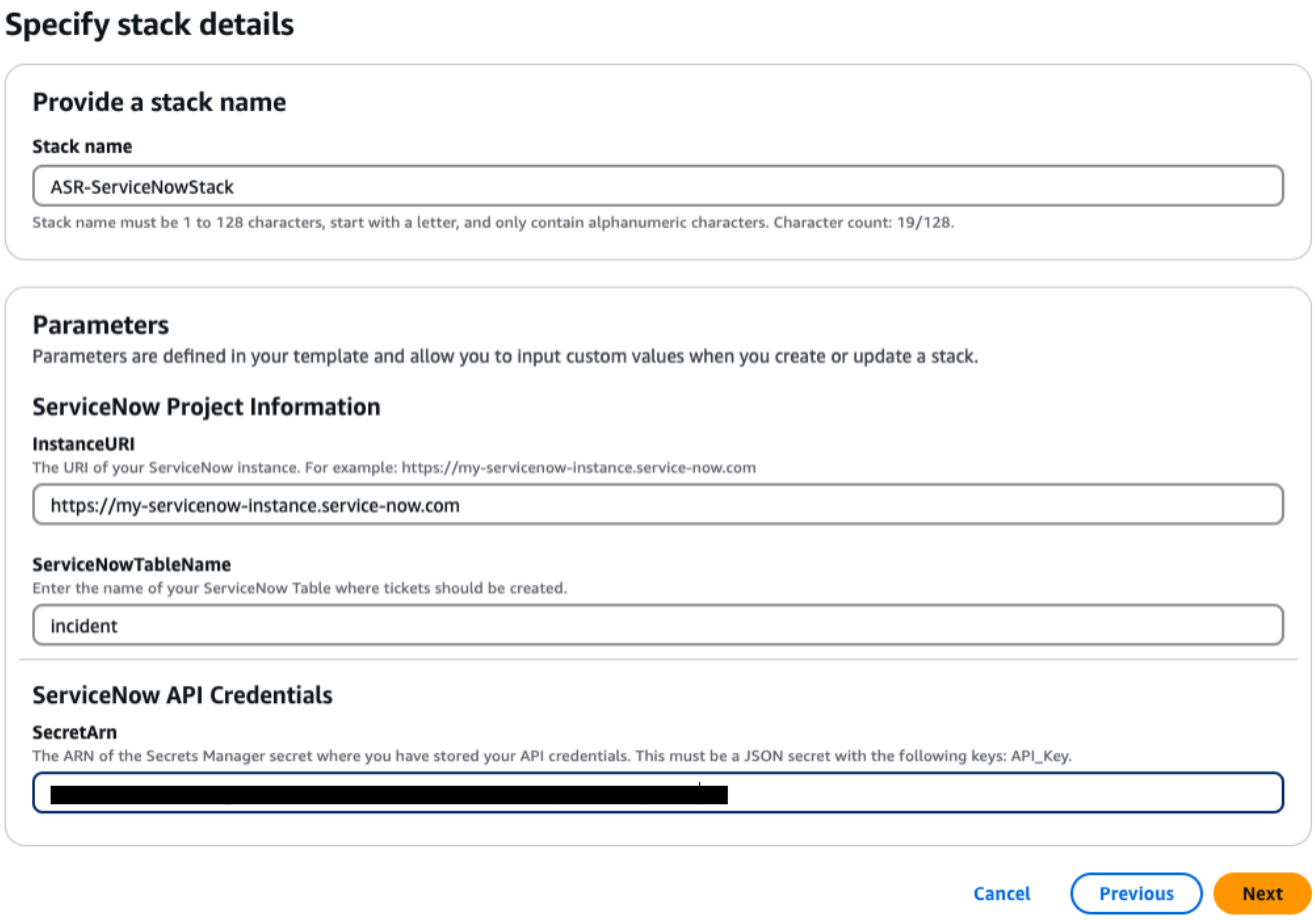
g. Provide the URI of your ServiceNow instance.

h. Provide your ServiceNow table name.

i. Create an API key in ServiceNow with permission to modify the table you intend to write to.

j. Create a secret in Secrets Manager with the key `API_Key` and provide the secret ARN as input to the stack.

**Provide a stack name ServiceNow project information, and ServiceNow API credentials.**




**To create a custom integration stack**: Include a Lambda function that the solution orchestrator Step Functions can call for each remediation. The Lambda function should take the input provided by Step Functions, construct a payload according to the requirements of your ticketing system, and make a request to your system to create the ticket.

## Step 1: Launch the admin stack

> **Important**
>
> This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered though this survey. Data collection is subject to the AWS Privacy Notice.
>
> To opt out of this feature, download the template, modify the AWS CloudFormation mapping section, and then use the AWS CloudFormation console to upload your template

and deploy the solution. For more information, refer to the Anonymized data collection section of this guide.

This automated AWS CloudFormation template deploys the Automated Security Response on AWS solution in the AWS Cloud. Before you launch the stack, you must enable Security Hub and complete the prerequisites.

**Note**

You are responsible for the cost of the AWS services used while running this solution. For more details, visit to the Cost section in this guide, and refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console from the account where the AWS Security Hub is currently configured, and use the button below to launch the `automated-security-response-admin.template` AWS CloudFormation template.


   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 this solution in a different AWS Region, use the Region selector in the AWS Management Console navigation bar.

   **Note**

   This solution uses AWS Systems Manager which is currently available in specific AWS Regions only. The solution works in all of the Regions that support this service. For the most current availability by Region, refer to the AWS Regional Services List.

3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and then choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and STS limits in the *AWS Identity and Access Management User Guide*.

5. On the **Parameters** page, choose **Next**.

| Parameter | Default | Description |
| --- | --- | --- |
| **Load SC Admin Stack** | yes | Specify whether to install the admin components for automated remediation of SC controls. |
| **Load AFSBP Admin Stack** | no | Specify whether to install the admin components for automated remediation of FSBP controls. |
| **Load CIS120 Admin Stack** | no | Specify whether to install the admin components for automated remediation of CIS120 controls. |
| **Load CIS140 Admin Stack** | no | Specify whether to install the admin components for automated remediation of CIS140 controls. |
| **Load CIS300 Admin Stack** | no | Specify whether to install the admin components for automated remediation of CIS300 controls. |
| **Load PC1321 Admin Stack** | no | Specify whether to install the admin components for automated remediation of PC1321 controls. |
| **Load NIST Admin Stack** | no | Specify whether to install the admin components for automated remediation of NIST controls. |

| Parameter | Default | Description |
| --- | --- | --- |
| **Reuse Orchestrator Log Group** | no | Select whether or not to reuse an existing `SO0111-ASR-Orchestrator` CloudWatch Logs group. This simplifies reinstallation and upgrades without losing log data from a previous version. Reuse existing `Orchestrator Log Group` choose `yes` if the `Orchestrator Log Group` still exists from an earlier deployment in this account, otherwise no. If you are performing a stack update from an earlier version than v2.3.0 choose no |
| **Use CloudWatch Metrics** | `yes` | Specify whether to enable CloudWatch Metrics for monitoring the solution. This will create a CloudWatch Dashboard for viewing metrics. |
| **Use CloudWatch Metrics Alarms** | `yes` | Specify whether to enable CloudWatch Metrics Alarms for the solution. This will create Alarms for certain metrics collected by the solution. |

| Parameter | Default | Description |
| --- | --- | --- |
| **RemediationFailureAlarmThreshold** | 5 | Specify the threshold for percentage of remediation failures per control ID. For example, if you enter 5, you receive an alarm if a control ID fails more than 5% of remediations at a given day.<br><br>This parameter functions only if alarms are created (see the **Use CloudWatch Metrics Alarms** parameter). |
| **EnableEnhancedCloudWatchMetrics** | no | If `yes`, creates additional CloudWatch metrics to track all control IDs individually on the CloudWatch dashboard and as CloudWatch alarms.<br><br>See the Cost section to understand the additional cost that this incurs. |
| **TicketGenFunctionName** | *(Optional input)* | Optional. Leave blank if you don't want to integrate a ticketing system. Otherwise, provide the Lambda function name from the stack output of Step 0, for example: `SO0111-ASR-ServiceNow-TicketGenerator` . |

| Parameter | Default | Description |
| --- | --- | --- |
| **TargetAccountIDs** | ALL | A list of AWS account IDs to control the scope of automated remediation.<br><br>Use "ALL" to target all accounts in the organization.<br><br>Or provide a comma-separated list of 12-digit AWS Account IDs. Example: "123456789012,098765432109" |
| **TargetAccountIDsStrategy** | INCLUDE | Defines how the solution applies automated remediations based on the TargetAccountIDs list.<br><br>INCLUDE: Run automated remediations only for accounts listed.<br><br>EXCLUDE: Run automated remediations for all accounts except those listed. |

> **Note**
>
> You must manually enable automatic remediations in the Admin account after deploying or updating the solution's CloudFormation stacks.

1. On the **Configure stack options** page, choose **Next**.
2. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.

3. 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 receive a CREATE_COMPLETE status in approximately 15 minutes.

## Step 2: Install the remediation roles into each AWS Security Hub member account

The `automated-security-response-member-roles.template` StackSet must be deployed in only one Region per member account. It defines the global roles that allow cross-account API calls from the ASR Orchestrator step function.

1. Sign in to the AWS Management Console for each AWS Security Hub member account (including the admin account, which is also a member). Select the button to launch the `automated-security-response-member-roles.template` AWS CloudFormation template. 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 this solution in a different AWS Region, use the Region selector in the AWS Management Console navigation bar.
3. On the **Create stack** page, verify that the correct template URL is in the Amazon S3 URL text box and then choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and STS limits in the AWS Identity and Access Management User Guide.
5. On the **Parameters** page, specify the following parameters and choose Next.

| Parameter | Default | Description |
| --- | --- | --- |
| **Namespace** | *<Requires input>* | Enter a string of up to 9 lowercase alphanumeric characters. Unique namespace to be added as a suffix to remediation IAM role names. The same |

| Parameter | Default | Description |
| --- | --- | --- |
| | | namespace should be used in the Member Roles and Member stacks. This string should be unique for each solution deployment, but does not need to be changed during stack updates. The namespace value does **not** need to be unique per member account. |
| **Sec Hub Account Admin** | *<Requires input>* | Enter the 12-digit account ID for the AWS Security Hub admin account. This value grants permissions to the admin account's solution role. |

6. On the **Configure stack options** page, choose **Next**.
7. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
8. 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 receive a CREATE_COMPLETE status in approximately 5 minutes. You may continue with the next step while this stack loads.

## Step 3: Launch the member stack

> **Important**
>
> This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered though this survey. Data collection is subject to the AWS Privacy Policy.

To opt out of this feature, download the template, modify the AWS CloudFormation mapping section, and then use the AWS CloudFormation console to upload your template and deploy the solution. For more information, refer to the Collection of operational metrics section of this guide.

The `automated-security-response-member` stack must be installed into each Security Hub member account. This stack defines the runbooks for automated remediation. The admin for each member account can control what remediations are available via this stack.

1. Sign in to the AWS Management Console for each AWS Security Hub member account (including the admin account, which is also a member). Select the button to launch the `automated-security-response-member.template` AWS CloudFormation template.


You can also download the template as a starting point for your own implementation. . The template launches in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the Region selector in the AWS Management Console navigation bar.

+

**Note**

This solution uses AWS Systems Manager, which is currently available in the majority of AWS Regions. The solution works in all of the Regions that support these services. For the most current availability by Region, refer to the AWS Regional Services List.

1. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and then choose **Next**.
2. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and STS limits in the *AWS Identity and Access Management User Guide*.
3. On the **Parameters** page, specify the following parameters and choose **Next**.

| Parameter | Default | Description |
| --- | --- | --- |
| **Provide the name of the LogGroup to be used to create Metric Filters and Alarms** | *<Requires input>* | Specify the name of a CloudWatch Logs group where CloudTrail logs API calls. This is used for CIS 3.1-3.14 remediations. |
| **Load SC Member Stack** | yes | Specify whether to install the member components for automated remediation of SC controls. |
| **Load AFSBP Member Stack** | no | Specify whether to install the member components for automated remediation of FSBP controls. |
| **Load CIS120 Member Stack** | no | Specify whether to install the member components for automated remediation of CIS120 controls. |
| **Load CIS140 Member Stack** | no | Specify whether to install the member components for automated remediation of CIS140 controls. |
| **Load CIS300 Member Stack** | no | Specify whether to install the member components for automated remediation of CIS300 controls. |
| **Load PC1321 Member Stack** | no | Specify whether to install the member components for automated remediation of PC1321 controls. |

| Parameter | Default | Description |
| --- | --- | --- |
| **Load NIST Member Stack** | no | Specify whether to install the member components for automated remediation of NIST controls. |
| **Create S3 Bucket For Redshift Audit Logging** | no | Select `yes` if the S3 bucket should be created for the FSBP RedShift.4 remediation. For details of the S3 bucket and the remediation, review the Redshift.4 remediation in the *AWS Security Hub User Guide*. |
| **Sec Hub Admin Account** | *<Requires input>* | Enter the 12-digit account ID for the AWS Security Hub admin account. |
| **Namespace** | *<Requires input>* | Enter a string of up to 9 lowercase alphanumeric characters. This string becomes part of the IAM role names and Action Log S3 bucket. Use the same value for member stack deployment and member roles stack deployment. String should be unique for each solution deployment, but does not need to be changed during stack updates. |

| Parameter | Default | Description |
| --- | --- | --- |
| **EnableCloudTrailForASRActionLog** | no | Select yes if you want to monitor management events conducted by the solution on the CloudWatch dashboard. The solution creates a CloudTrail trail in each member account where you select yes. You must deploy the solution into an AWS Organization to enable this feature. See the Cost section to understand the additional cost that this incurs. |

4. On the **Configure stack options** page, choose **Next**.
5. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
6. Choose **Create stack** to deploy the stack.

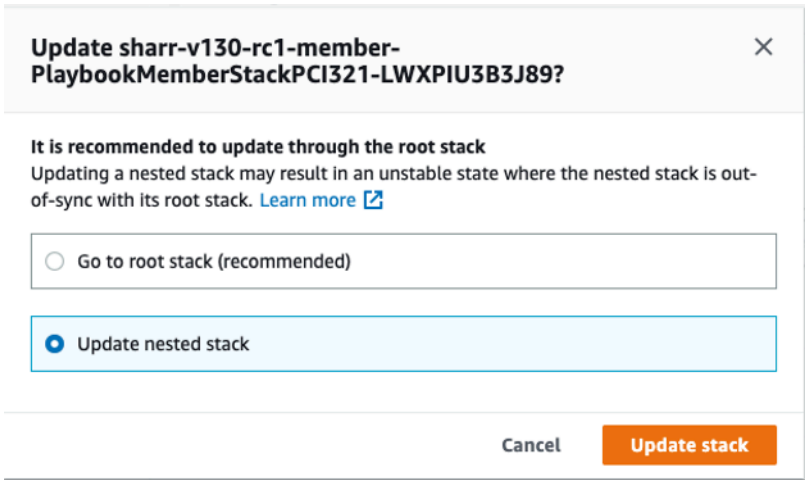
You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a CREATE_COMPLETE status in approximately 15 minutes.

## Step 4: (Optional) Adjust the available remediations

If you want to remove specific remediations from a member account, you can do so by updating the nested stack for the security standard. For simplicity, the nested stack options are not propagated to the root stack.

1. Sign in to the AWS CloudFormation console and select the nested stack.
2. Choose **Update**.
3. Select **Update nested stack** and choose **Update stack**.

**Update nested stack**




4. Select **Use current template** and choose **Next**.
5. Adjust the available remediations. Change the values for desired controls to `Available` and undesired controls to `Not available`.

> **Note**
>
> Turning off a remediation removes the solutions remediation runbook for the security standard and control.

6. On the **Configure stack options** page, choose **Next**.
7. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
8. Choose **Update stack**.

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

# Control Tower (CT) deployment

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

**Time to deploy:** Approximately 30 minutes

## Prerequisites

Before deploying this solution, ensure that it is intended for **AWS Control Tower administrators**.

When you're ready to set up your landing zone using the AWS Control Tower console or APIs, follow these steps:

To get started with AWS Control Tower, see: Getting Started with AWS Control Tower

To learn how to customize your landing zone, refer to: Customizing Your Landing Zone

To launch and deploy your landing zone, see: Landing Zone Deployment Guide

## Deployment overview

Use the following steps to deploy this solution on AWS.

Step 1: Build and deploy S3 bucket

> **Note**
>
> S3 bucket Configuration – for ADMIN only. This is a one-time setup step and should not be repeated by end users. The S3 buckets store the deployment package, including the AWS CloudFormation template and Lambda code required for ASR to run. These resources are deployed using CfCt or StackSet.

**1. Configure the S3 Bucket**

Set up the S3 bucket that will be used for storing and serving your deployment packages.

**2. Set Up the Environment**

Prepare the necessary environment variables, credentials, and tools required for the build and deployment process.

**3. Configure S3 Bucket Policies**

Define and apply the appropriate bucket policies to control access and permissions.

**4. Prepare the Build**

Compile, package, or otherwise prepare your application or assets for deployment.

**5. Deploy Packages to S3**

Upload the prepared build artifacts to the designated S3 bucket.

Step 2: Stacks deployment to AWS Control Tower

**1. Create Build Manifest for ASR Components**

Define a build manifest that lists all ASR components, their versions, dependencies, and build instructions.

**2. Update the CodePipeline**

Modify the AWS CodePipeline configuration to include the new build steps, artifacts, or stages required for deploying the ASR components.

## Step 1: Build and deploy to S3 bucket

AWS Solutions use two buckets: a bucket for global access to templates, which is accessed via HTTPS, and regional buckets for access to assets within the region, such as Lambda code.

**1. Configure the S3 Bucket**

Pick a unique bucket name, e.g. asr-staging. Set two environment variables on your terminal, one should be the base bucket name with -reference as suffix, the other with your intended deployment region as suffix:

```
export BASE_BUCKET_NAME=asr-staging-$(date +%s)
export TEMPLATE_BUCKET_NAME=$BASE_BUCKET_NAME-reference
export REGION=us-east-1
export ASSET_BUCKET_NAME=$BASE_BUCKET_NAME-$REGION
```

**2. Environment Setup**

In your AWS account, create two buckets with these names, e.g. asr-staging-reference and asr-staging-us-east-1. (The reference bucket will hold the CloudFormation templates, the regional

bucket will hold all other assets like the lambda code bundle.) Your buckets should be encrypted and disallow public access

```
aws s3 mb s3://$TEMPLATE_BUCKET_NAME/
aws s3 mb s3://$ASSET_BUCKET_NAME/
```

> **Note**
>
> When creating your buckets, ensure they are not publicly accessible. Use random bucket names. Disable public access. Use KMS encryption. And verify bucket ownership before uploading.

### 3. S3 buckets policy setup

Update the $TEMPLATE_BUCKET_NAME S3 bucket policy to include PutObject permissions for the execute account ID. Assign this permission to an IAM role within the execute account that is authorized to write to the bucket. This setup allows you to avoid creating the bucket in the Management account.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::<template bucket name>/*",
                "arn:aws:s3:::<template bucket name>"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "<org id>"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:PutObject",
```

```
            "Resource": [
                "arn:aws:s3:::<template bucket name>/*",
                "arn:aws:s3:::<template bucket name>"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:PrincipalArn": "arn:aws:iam::<execute_account_id>:role/
<iam_role_name>"
                }
            }
        }
    ]
}
```

Alter the asset S3 bucket policy to include permissions. Assign this permission to an IAM role within the execute account that is authorized to write to the bucket. Repeat this setup for each regional asset bucket (e.g., asr-staging-us-east-1, asr-staging-eu-west-1, etc.), allowing deployments across multiple regions without needing to create the buckets in the Management account.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::<asset bucket name>-<region>/*",
                "arn:aws:s3:::<asset bucket name>-<region>"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "<org id>"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": [
            "arn:aws:s3:::<asset bucket name>-<region>/*",
                "arn:aws:s3:::<asset bucket name>-<region>"
```

```
            ],
            "Condition": {
                "ArnLike": {
                    "aws:PrincipalArn": "arn:aws:iam::<execute_account_id>:role/
<iam_role_name>"
                }
            }
        }
    ]
}
```

### 4. Build Preparation

- Prerequisites:
  - AWS CLI v2
  - Python 3.11+ with pip
  - AWS CDK 2.171.1+
  - Node.js 20+ with npm
  - Poetry v2 with plugin to export
- Git clone https://github.com/aws-solutions/automated-security-response-on-aws.git

First ensure that you've run npm install in the source folder.

Next from the deployment folder in your cloned repo, run build-s3-dist.sh, passing the root name of your bucket (ex. mybucket) and the version you are building (ex. v1.0.0). We recommend using a semver version based on the version downloaded from GitHub (ex. GitHub: v1.0.0, your build: v1.0.0.mybuild)

```
chmod +x build-s3-dist.sh
export SOLUTION_NAME=automated-security-response-on-aws
export SOLUTION_VERSION=v1.0.0.mybuild
./build-s3-dist.sh -b $BASE_BUCKET_NAME -v $SOLUTION_VERSION
```

### 5. Deploy packages to S3

```
cd deployment
aws s3 cp global-s3-assets/  s3://$TEMPLATE_BUCKET_NAME/$SOLUTION_NAME/
$SOLUTION_VERSION/ --recursive --acl bucket-owner-full-control
```

```
aws s3 cp regional-s3-assets/  s3://$ASSET_BUCKET_NAME/$SOLUTION_NAME/
$SOLUTION_VERSION/ --recursive --acl bucket-owner-full-control
```

## Step 2: Stacks deployment to AWS Control Tower

### 1. Build manifest for ASR components

After deploying ASR artifacts to the S3 buckets, update the Control Tower pipeline manifest to reference the new version, and then trigger the pipeline run, refer to: controltower deployment

> **Important**
>
> To ensure correct deployment of the ASR solution, refer to the official AWS documentation for detailed information on the CloudFormation templates overview and parameters description. Info links below: CloudFormation Templates Parameters overview Guide

The manifest for the ASR components looks like this:

```
region: us-east-1 #<HOME_REGION_NAME>
version: 2021-03-15

# Control Tower Custom CloudFormation Resources
resources:
  - name: <ADMIN STACK NAME>
    resource_file: s3://<ADMIN TEMPLATE BUCKET path>
    parameters:
      - parameter_key: UseCloudWatchMetricsAlarms
        parameter_value: "yes"
      - parameter_key: TicketGenFunctionName
        parameter_value: ""
      - parameter_key: LoadSCAdminStack
        parameter_value: "yes"
      - parameter_key: LoadCIS120AdminStack
        parameter_value: "no"
      - parameter_key: TargetAccountIDsStrategy
        parameter_value: "INCLUDE"
      - parameter_key: LoadCIS300AdminStack
        parameter_value: "no"
      - parameter_key: UseCloudWatchMetrics
        parameter_value: "yes"
      - parameter_key: LoadNIST80053AdminStack
```

```
        parameter_value: "no"
      - parameter_key: LoadCIS140AdminStack
        parameter_value: "no"
      - parameter_key: ReuseOrchestratorLogGroup
        parameter_value: "yes"
      - parameter_key: LoadPCI321AdminStack
        parameter_value: "no"
      - parameter_key: RemediationFailureAlarmThreshold
        parameter_value: "5"
      - parameter_key: LoadAFSBPAdminStack
        parameter_value: "no"
      - parameter_key: TargetAccountIDs
        parameter_value: "ALL"
      - parameter_key: EnableEnhancedCloudWatchMetrics
        parameter_value: "no"
    deploy_method: stack_set
    deployment_targets:
     accounts: # :type: list
        - <ACCOUNT_NAME> # and/or
        - <ACCOUNT_NUMBER>
    regions:
      - <REGION_NAME>

  - name:  <ROLE MEMBER STACK NAME>
    resource_file: s3://<ROLE MEMBER TEMPLATE BUCKET path>
    parameters:
      - parameter_key: SecHubAdminAccount
        parameter_value: <ADMIN_ACCOUNT_NAME>
      - parameter_key: Namespace
        parameter_value: <NAMESPACE>
    deploy_method: stack_set
    deployment_targets:
      organizational_units:
        - <ORG UNIT>

  - name:  <MEMBER STACK NAME>
    resource_file: s3://<MEMBER TEMPLATE BUCKET path>
    parameters:
      - parameter_key: SecHubAdminAccount
        parameter_value: <ADMIN_ACCOUNT_NAME>
      - parameter_key: LoadCIS120MemberStack
        parameter_value: "no"
      - parameter_key: LoadNIST80053MemberStack
        parameter_value: "no"
```

```
    - parameter_key: Namespace
      parameter_value: <NAMESPACE>
    - parameter_key: CreateS3BucketForRedshiftAuditLogging
      parameter_value: "no"
    - parameter_key: LoadAFSBPMemberStack
      parameter_value: "no"
    - parameter_key: LoadSCMemberStack
      parameter_value: "yes"
    - parameter_key: LoadPCI321MemberStack
      parameter_value: "no"
    - parameter_key: LoadCIS140MemberStack
      parameter_value: "no"
    - parameter_key: EnableCloudTrailForASRActionLog
      parameter_value: "no"
    - parameter_key: LogGroupName
      parameter_value: <LOG_GROUP_NAME>
    - parameter_key: LoadCIS300MemberStack
      parameter_value: "no"
  deploy_method: stack_set
  deployment_targets:
   accounts: # :type: list
      - <ACCOUNT_NAME> # and/or
      - <ACCOUNT_NUMBER>
  organizational_units:
    - <ORG UNIT>
  regions: # :type: list
    - <REGION_NAME>
```

### 2. Code pipeline update

Add a manifest file to a custom-control-tower-configuration.zip and run a CodePipeline, refer to: code pipeline overview

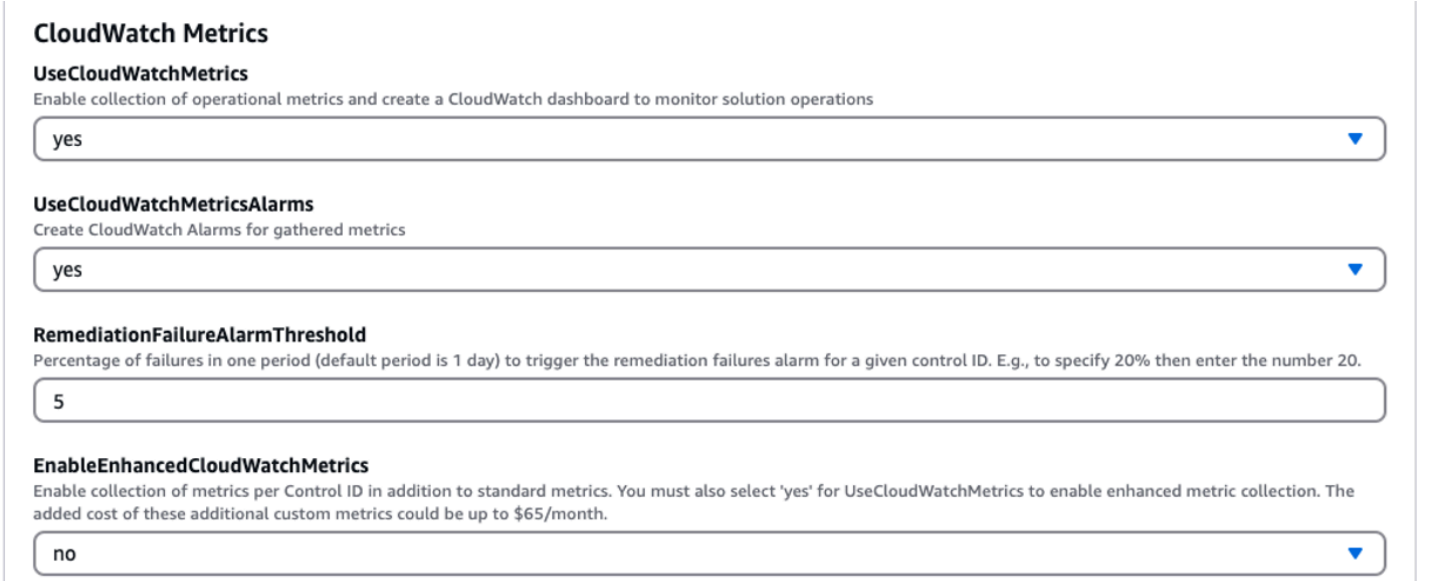
# Monitor the solution's operations with an Amazon CloudWatch dashboard

This solution includes custom metrics and alarms displayed on an Amazon CloudWatch dashboard.

The CloudWatch dashboard and alarms monitor the solution’s operations and alerts when there is a potential issue.

## Enabling CloudWatch metrics, alarms, and dashboard

There are four CloudFormation template parameters for CloudWatch functionality.




1. **UseCloudWatchMetrics** - Setting this to `yes` enables the collection of operational metrics and creates a CloudWatch dashboard to view these metrics.
2. **UseCloudWatchAlarms** - Setting this to `yes` enables the solution’s default alarms.
3. **RemediationFailureAlarmThreshold** - The percentage of failing remediations in a period to raise an alarm.
4. **EnableEnhancedCloudWatchMetrics** - Set this parameter to `yes` to collect individual metrics per control ID. By default, this parameter is set to `no`, so that only metrics on the total number of remediations across all control IDs are collected. Individual metrics and alarms per control ID incur additional cost.

# Using the CloudWatch dashboard

To view the dashboard:

1. Navigate to Amazon CloudWatch and then Dashboards.
2. Select the dashboard named "ASR-Remediation-Metrics-Dashboard".

The CloudWatch dashboard contains the following sections:

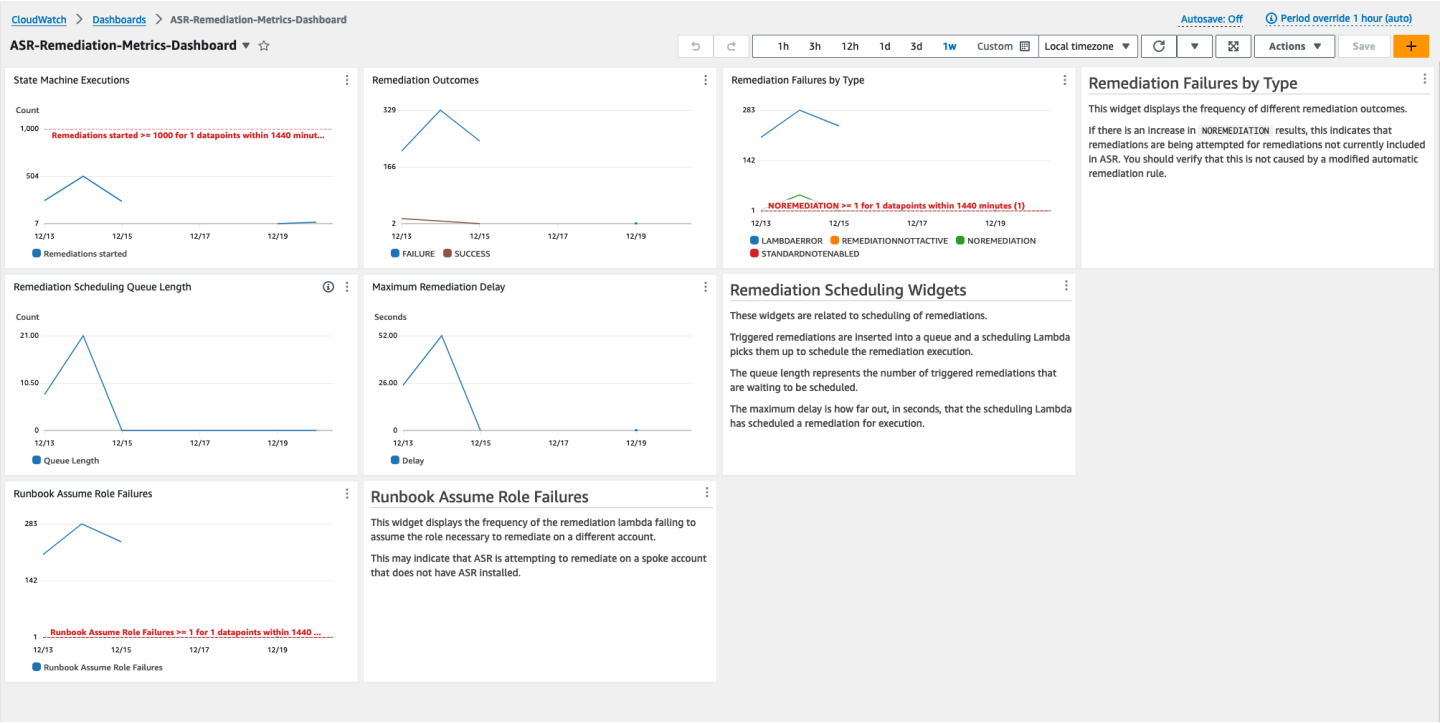
1. **Total Successful Remediations** - Gives you insight into the number of Security Hub findings that have been successfully remediated by the solution.
2. **Remediation Failures** - Shows how many remediations have been failing, both in total and as a percentage, and the failure cause. A high number of failures can hint at a technical problem with the solution that you might need to investigate in more detail.
3. **Remediation Success/Failure by Control ID** - If you enabled Enhanced Metrics at deployment time, this section lists remediation results by control ID. When the **Remediation Failures** section shows a high failure rate in general, this section shows you whether the failures are distributed across many control IDs, or if only certain control IDs are failing.
4. **Runbook Assume Role Failures** - Shows the number of failures that occurred because of remediation attempts in accounts that don't have the solution Member role installed. Repeated failures by automated remediation attempts due to missing roles cause unnecessary cost. Mitigate this by installing the Member role stack in the concerned accounts, disabling all EventBridge rules created by the solution, or disassociating the account in Security Hub.
5. **Cloud Trail Management Actions by ASR** - Lists management actions by the solution across all member accounts where you enabled Action Logs with the **EnableCloudTrailForASRActionLog** parameter at deployment time. When you observe unexpected resource changes in any of your AWS accounts, this widget can help you understand if resources were modified by the solution.

The CloudWatch dashboard also comes with predefined alarms that alert to common operational errors.

1. State Machine executions > 1000 in a 24-hour period.
   a. A large spike in remediation executions could indicate an event rule is initiating more often than intended.
   b. Threshold can be changed using the CloudFormation parameter.

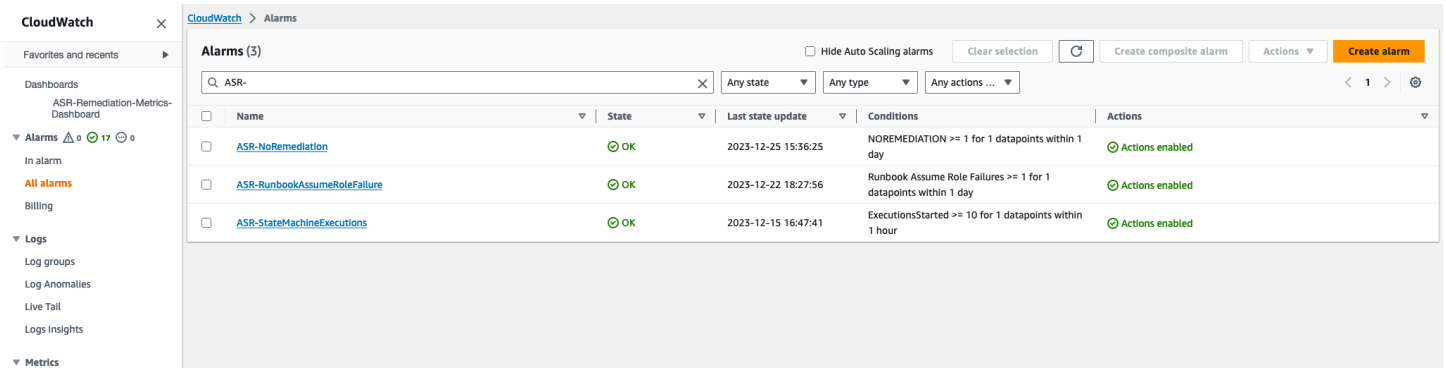
2. Remediation Failures by Type = NOREMEDIATION > 0
    a. Remediations are being attempted for remediations that are not included in ASR. This could indicate an event rule has been modified to include more than the intended remediations.
3. Runbook Assume Role Failures > 0
    a. Remediations are being attempted on accounts or Regions that do not have the solution properly deployed. This could indicate an event rule has been modified to include more accounts than intended.

All alarm thresholds can be modified to suit the individual deployment needs.

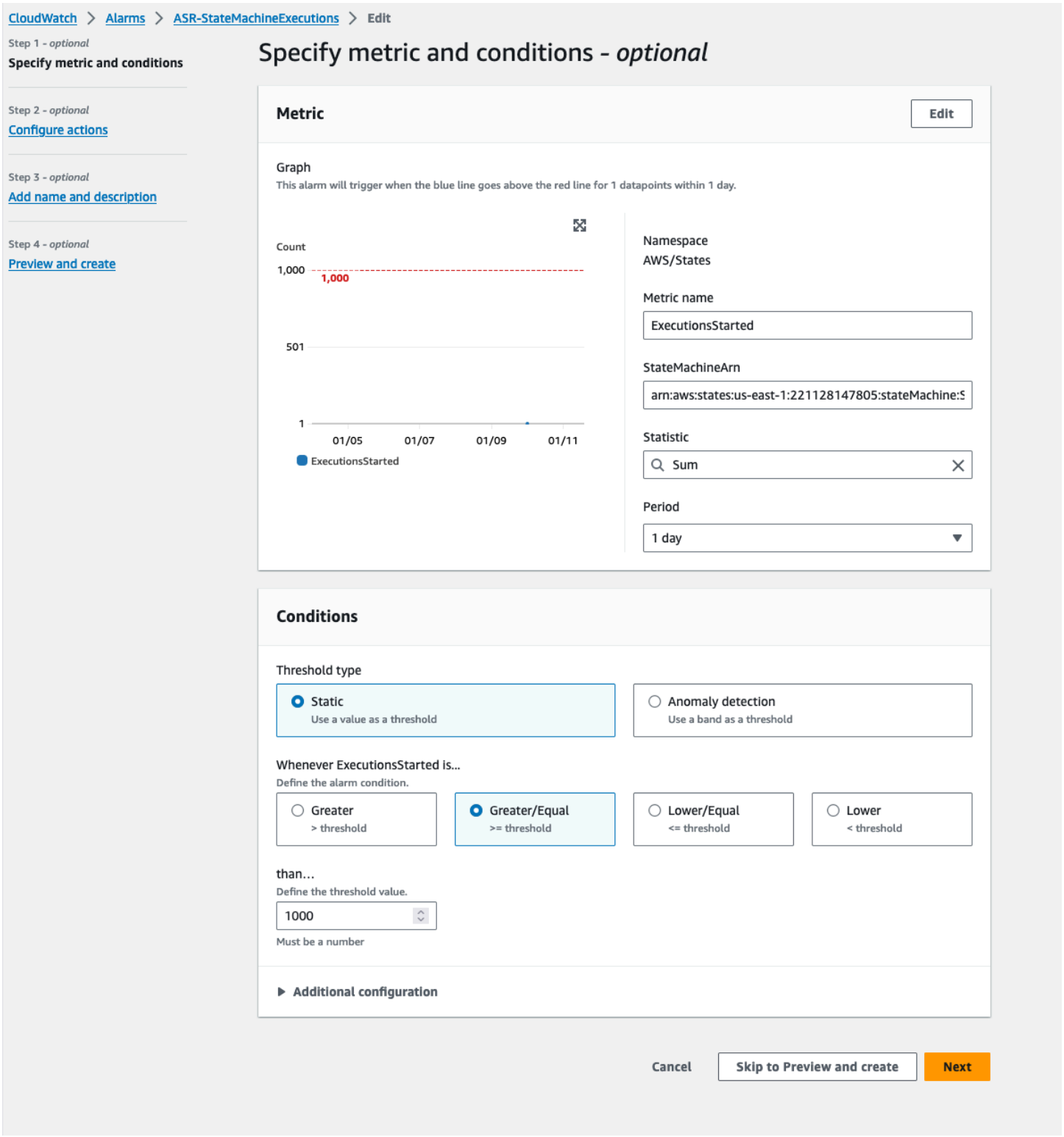



# Modifying alarm thresholds

1. Navigate to Amazon CloudWatch → Alarms → All Alarms.
2. Choose the Alarm you would like to modify, then select Actions → Edit.



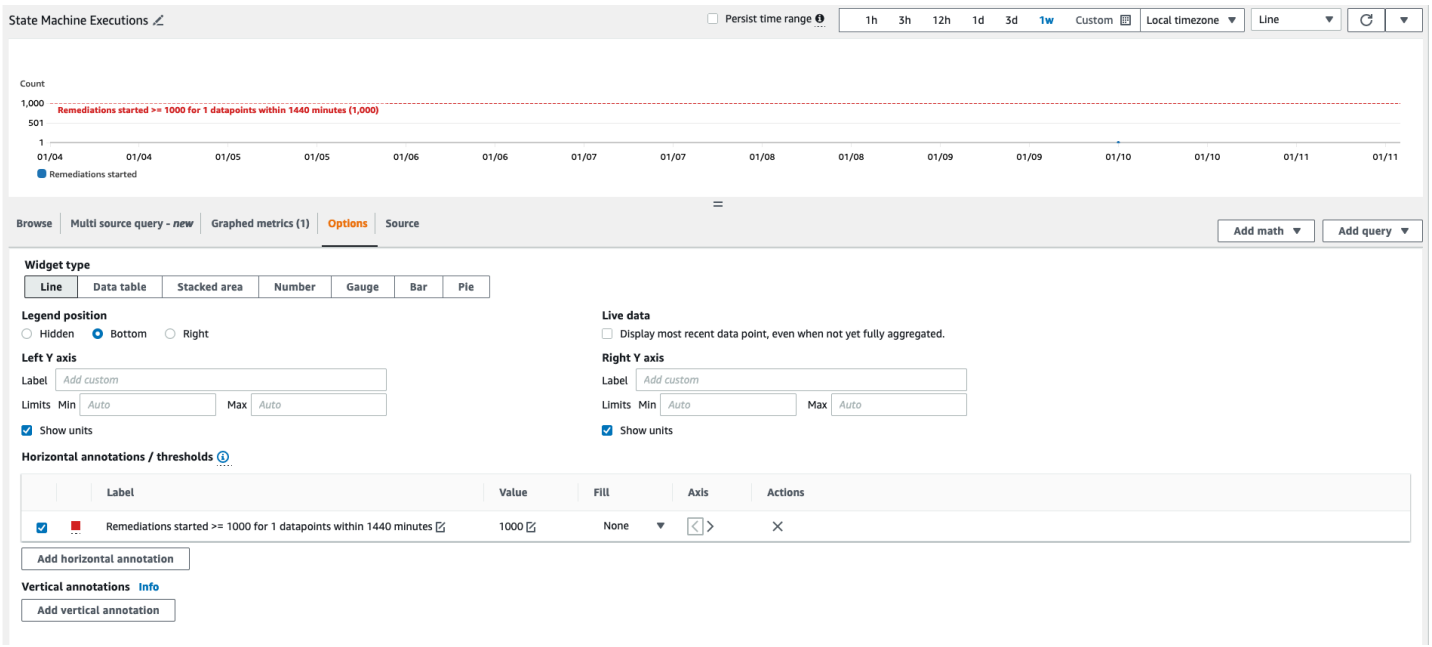

1. Change the threshold to the desired value and save.




1. Navigate to the CloudWatch dashboard to modify the charts there to match the new settings.
   a. Select the ellipsis on the top right of the corresponding widget.
   b. Select Edit.

c. Change to the Options tab.

d. Modify the Alarm annotation to match the new settings.




# Subscribing to Alarm notifications

In the admin account, subscribe to the Amazon SNS topic created by the admin stack, SO0111-ASR_Alarm_Topic. This will notify you when an alarm enters the ALARM state.

# Update the solution

## Upgrading from versions prior to v1.4

If you have previously deployed the solution prior to v1.4.x, uninstall, then install the latest version:

1. Uninstall the previously deployed solution. Refer to Uninstall the solution.
2. Launch the latest template. Refer to Deploy the solution.

> **Note**
>
> If you are upgrading from v1.2.1 or earlier to v1.3.0 or later, set **Use existing Orchestrator Log Group** to No. If you are reinstalling v1.3.0 or later, you can select Yes for this option. This option allows you to continue to log to the same Log Group for the Orchestrator Step Functions.

## Upgrading from v1.4 and later

If you are upgrading from v1.4.x, update all stacks or StackSets as follows:

1. Update the stack in the Security Hub admin account using the latest template.
2. In each member account, update the permissions from the latest template.
3. In each member account in all Regions where currently deployed, update the member stack from the latest template.

## Upgrading from v2.0.x

If you are upgrading from v2.0.x, upgrade to v2.1.2 or later. Updating to v2.1.0 - v2.1.1 will fail in CloudFormation.

> **Note**
>
> - When updating the solution, automated remediation rules may need to be re-enabled manually in the Admin account. Refer to Enable fully-automated remediations.

- If you are using the `Reuse Orchestrator Log Group` parameter to retain logs, ensure it is set appropriately during stack update to avoid log group recreation or loss of log retention settings. Refer to Deploy the solution. If you are performing a stack update to v2.3.0+ from an earlier version choose "no"

# Troubleshooting

Known issue resolution provides instructions to mitigate known errors. If these instructions don't address your issue, Contact AWS Support provides instructions for opening an AWS Support case for this solution.

## Solution logs

This section includes Troubleshooting information for this solution, see left navigation for topics.

This solution collects output from remediation runbooks, which run under AWS Systems Manager, and logs the result to CloudWatch Logs group `SO0111-ASR` in the AWS Security Hub admin account. There is one stream per control per day.

The Orchestrator Step Functions logs all step transitions to the `SO0111-ASR-Orchestrator` CloudWatch Logs Group in the AWS Security Hub admin account. This log is an audit trail to record state transitions for each instance of the Step Functions. There is one log stream per Step Functions execution.

Both log groups are encrypted using an AWS KMS Customer-Manager Key (CMK).

The following troubleshooting information uses the `SO0111-ASR` log group. Use this log, as well as AWS Systems Manager Automation console, Automation Executions logs, Step Function console, and Lambda logs to troubleshoot problems.

If a remediation fails, a message similar to the following will be logged to `SO0111-ASR` in the log stream for the standard, control, and date. For example: **CIS-2.9-2021-08-12**

```
ERROR: a4cbb9bb-24cc-492b-a30f-1123b407a6253: Remediation failed for CIS control
 2.9 in account 123412341234: See Automation Execution output for details (AwsEc2Vpc
 vpc-0e92bbe911cf08acb)
```

The following messages provide additional detail. This output is from the ASR runbook for the security standard and control. For example: **ASR-CIS_1.2.0_2.9**

```
Step fails when it is Execution complete: verified. Failed to run automation with
 executionId: eecdef79-9111-4532-921a-e098549f5259 Failed :
```

```
{Status=[Failed], Output=[No output available yet because the step is not successfully
 executed], ExecutionId=[eecdef79-9111-4532-921a-e098549f5259]}. Please refer to
 Automation Service Troubleshooting Guide for more diagnosis details.
```

This information points you to the failure, which in this case was a child automation running in the member account. To troubleshoot this issue, you must log in to the AWS Management Console in the member account (from the message above), go to AWS Systems Manager, navigate to **Automation**, and examine the log output for Execution ID `eecdef79-9111-4532-921a-e098549f525`.

## Known issue resolution

- **Issue:** The solution deployment fails with an error stating that the resources are already available in Amazon CloudWatch.

  **Resolution:** Check for an error message in the CloudFormation resources/events section indicating log groups already exist. The ASR deployment templates allow reuse of existing log groups. Verify that you have selected reuse.

- **Issue:** Solution fails to deploy with an error in a playbook nested stack where an EventBridge Rule fails to create

  **Resolution:** You have likely hit the quota for EventBridge rules with the number of playbooks deployed. You can avoid this by using Consolidated control findings in Security Hub paired with the SC playbook in this solution, deploy only the playbooks for the standards used, or requesting an increase to the EventBridge rules quota.

- **Issue:** I run Security Hub in multiple Regions in the same account. I want to deploy this solution in multiple Regions.

  **Resolution:** Deploy the admin stack in the same account and Region as your Security Hub admin. Install the member template into each account and Region where you have a Security Hub member configured. Enable aggregation in the Security Hub.

- **Issue:** Immediately after deploying, the **SO0111-ASR-Orchestrator** is failing in the Get Automation Document State with a 502 error: *"`Lambda was unable to decrypt the environment variables because KMS access was denied. Please check the function's KMS key settings. KMS Exception: UnrecognizedClientExceptionKMS Message: The security token included in the request is invalid. (Service: AWSLambda; Status Code: 502; Error Code: KMSAccessDeniedException; Request ID: …`"*

**Resolution:** Allow the solution about 10 minutes to stabilize before running remediations. If the problem continues, open a support ticket or GitHub issue.

- **Issue**: I attempted to remediate a finding but nothing happened.

  **Resolution:** Check the notes of the finding for reasons why it was not remediated. A common cause is that the finding has no automated remediation. At this time there is no way to provide direct feedback to the user when no remediation exists other than via the notes. Review the solution logs. Open CloudWatch Logs in the console. Find the SO0111-ASR CloudWatch Logs Group. Sort the list so the most-recently updated streams appear first. Select the log stream for the finding you attempted to run. You should find any errors there. Some reasons for the failure could be: mismatch between finding control and remediation control, cross-account remediation (not yet supported), or that the finding has already been remediated. If unable to determine the reason for the failure, please collect the logs and open a support ticket.

- **Issue:** After starting a remediation, the status in the Security Hub console has not updated.

  **Resolution:** The Security Hub console does not update automatically. Refresh the current view. The status of the finding should update. It might take several hours for the finding to transition from **Failed** to **Passed**. Findings are created from event data sent by other services, such as AWS Config, to AWS Security Hub. The time before a rule is reevaluated depends on the underlying service. If this does not resolve the issue, refer to the preceding resolution for *"`I attempted to remediate a finding but nothing happened.`"*

- **Issue**: Orchestrator step function fails in **Get Automation Document State**: *An error occurred (AccessDenied) when calling the AssumeRole operation*.

  **Resolution**: The member template has not been installed in the member account where ASR is attempting to remediate a finding. Follow instructions for deployment of the member template.

- **Issue**: Config.1 runbook fails because Recorder or Delivery Channel already exists.

  **Resolution**: Inspect your AWS Config settings carefully to ensure Config is properly set up. The automated remediation is not able to fix existing AWS Config settings in some cases.

- **Issue**: Remediation is successful but returns the message `"No output available yet because the step is not successfully executed."`

  **Resolution**: This is a known issue in this release where certain remediation runbooks do not return a response. The remediation runbooks will properly fail and signal the solution if they do not work.

- **Issue**: The resolution failed and sent a stack trace.

  **Resolution**: Occasionally, we miss the opportunity to handle an error condition that results in a stack trace rather than an error message. Attempt to troubleshoot the problem from the trace data. Open a support ticket if you need assistance.
- **Issue**: Removal of the v1.3.0 stack failed on the Custom Action resource.

  **Resolution**: Removal of the admin template may fail on the Custom Action removal. This is a known issue that will be fixed in the next release. If this occurs:

  a. Sign in to AWS Security Hub management console.

  b. In the admin account, go to **Settings**.

  c. Select the **Custom actions** tab

  d. Manually delete the entry **Remediate with ASR**.

  e. Delete the stack again.
- **Issue**: After redeploying the admin stack the step function is failing on `AssumeRole`.

  **Resolution**: Redeploying the admin stack breaks the trust connection between the admin role in the admin account and the member role in the member accounts. You must redeploy the member roles stack in all member accounts.
- **Issue**: CIS 3.x remediations are not showing `PASSED` after more than 24 hours.

  **Resolution**: This is a common occurrence if you have no subscriptions to the `SO0111-ASR_LocalAlarmNotification` SNS topic in the member account.

# Issues with specific remediations

### SetSSLBucketPolicy fails with AccessDenied error

Associated controls: AWS FSBP v1.0.0 S3.5, PCI v3.2.1 PCI.S3.5, CIS v1.4.0 2.1.2, SC v2.0.0 S3.5

**Issue:** The SetSSLBucketPolicy fails with an AccessDenied error:

*An error occurred (AccessDenied) when calling the PutBucketPolicy operation: Access Denied*

If the Block Public Access setting has been enabled for a bucket, attempts to put a bucket policy that includes statements that allow public access with fail with this error. This state can be reached by putting a bucket policy that contains such statements, then enabling the public access block for that bucket.

The remediation ConfigureS3BucketPublicAccessBlock (associated controls: AWS FSBP v1.0.0 S3.2, PCI v3.2.1 PCI.S3.2, CIS v1.4.0 2.1.5.2, SC v2.0.0 S3.2) can also put a bucket into this state because it sets the public access block setting without changing the bucket policy.

The SetSSLBucketPolicy adds a statement to the bucket policy to deny requests that do not use SSL. It does not modify the other statements in the policy, so if there are statements that allow public access, the remediation will fail attempting to put the modified bucket polic that still includes those statements.

**Resolution:** Modify the bucket policy to remove statements that allow public access in conflict with the block public access setting on the bucket.

## PutS3BucketPolicyDeny fails

Associated controls: AWS FSBP v1.0.0 S3.6, NIST.800-53.r5 CA-9(1), NIST.800-53.r5 CM-2

Issue: The PutS3BucketPolicyDeny with the following error:

```
Unable to create an explicit deny statement for {bucket_name}.
```

If the principals for all policies on the target bucket are "*", the solution cannot add the deny policy to the target bucket as it would block out all bucket actions for all principals.

**Resolution:** Modify the bucket policy to allow actions to specific accounts instead of using "*" principals and restrict denied actions.

# How to disable the solution

In the event of an incident, you may find that you need to disable the solution without removing any of the infrastructure. These scenarios detail how to disable different components in the solution.

**Scenario 1**: Disable automatic remediation for a single control.

1. Navigate to EventBridge in the AWS CloudFormation console.
2. Select Rules in the sidebar.
3. Select the default event bus and search for the control that you would like to disable.
4. Select on the rule and select the Disable button.

**Scenario 2**: Disable automatic remediation for all controls.

1. Navigate to EventBridge in the console.
2. Select Rules in the sidebar.
3. Select the "default" event bus and select all rules below.
4. Select on the "Disable" button. Note that you may have to do this for multiple pages of rules.

**Scenario 3**: Disable manual remediation for an account

1. Navigate to EventBridge in the console.
2. Select Rules in the sidebar.
3. Select the "default" event bus and search for "Remediate_with_ASR_CustomAction"
4. Select on the rule and select the "Disable" button.

# Contact Support

If you have AWS Developer Support, AWS Business Support, or AWS Enterprise Support, you can use the Support Center to get expert assistance with this solution. The following sections provide instructions.

## Create case

1. Sign in to Support Center.
2. Choose **Create case**.

## How can we help?

1. Choose **Technical**.
2. For **Service**, select **Solutions**.
3. For **Category**, select **Other Solutions**.
4. For **Severity**, select the option that best matches your use case.
5. When you enter the **Service**, **Category**, and **Severity**, the interface populates links to common troubleshooting questions. If you can't resolve your question with these links, choose **Next step: Additional information**.

## Additional information

1. For **Subject**, enter text summarizing your question or issue.
2. For **Description**, describe the issue in detail.
3. Choose **Attach files**.
4. Attach the information that Support needs to process the request.

## Help us resolve your case faster

1. Enter the requested information.
2. Choose **Next step: Solve now or contact us**.

## Solve now or contact us

1. Review the **Solve now** solutions.
2. If you can't resolve your issue with these solutions, choose **Contact us**, enter the requested information, and choose **Submit**.

# Uninstall the solution

Use the following procedure to uninstall the solution with the AWS Management Console.

## V1.0.0-V1.2.1

For releases v1.0.0 to v1.2.1, use Service Catalog to uninstall the CIS and/or FSBP Playbooks. With v1.3.0 Service Catalog is no longer used.

1. Sign in to the AWS CloudFormation console and navigate to the Security Hub primary account.
2. Choose **Service Catalog** to terminate any provisioned playbooks, remove any security groups, roles, or users.
3. Remove the spoke `CISPermissions.template` template form the Security Hub member accounts.
4. Remove the spoke `AFSBPMemberStack.template` template form the Security Hub admin and member accounts.
5. Navigate to the Security Hub primary account, select the solution's installation stack, and then choose **Delete**.

> **Note**
>
> CloudWatch Logs group logs are retained. We recommend retaining these logs as required by your organization's log retention policy.

## V1.3.x

1. Remove the `automated-security-response-member.template` from each member account.
2. Remove the `automated-security-response-admin.template` from the admin account.

**Note**

Removal of the admin template in v1.3.0 will likely fail on the Custom Action removal. This is a known issue that will be fixed in the next release. Use the following instructions to fix this issue:

1. Sign in to the AWS Security Hub management console.
2. In the admin account, go to **Settings**.
3. Select the **Custom actions** tab.
4. Manually delete the entry **Remediate with ASR**.
5. Delete the stack again.

# V1.4.0 and later

### Stack deployment

1. Remove the `automated-security-response-member.template` from each member account.
2. Remove the `automated-security-response-admin.template` from the admin account.

### StackSet deployment

For each StackSet, remove stacks, then remove the StackSet in the reverse order of deployment.

Note that IAM roles from the `automated-security-response-member-roles.template` are retained even if the template is removed. This is so that remediations using these roles continue to function. These SO0111-* roles can be manually removed after verifying that they are no longer in use by active remediations, such as CloudTrail to CloudWatch logging, or RDS Enhanced Monitoring.

# Administrator guide

## Enabling and disabling parts of the solution

As a solution administrator, you have the following controls over which functionalities of the solution are enabled.

**Where the member and member roles stacks are deployed:**

- The admin stack will only be able to initiate remediations (through custom action or fully automated EventBridge rules) in accounts in which the member and member roles stacks have been deployed with the admin account number given as a parameter value.
- To exempt accounts or Regions from control of the solution completely, do not deploy the member or member roles stacks to those accounts or Regions.

**Account and Region finding aggregation configuration in Security Hub:**

- The admin stack will only be able to initiate remediations (through custom action or fully automated EventBridge rules) for findings which arrive in the admin account and Region.
- To exempt accounts or Regions from control of the solution completely, do not include those accounts or Regions to send findings to the same admin account and Region in which the admin stack is deployed.

**Which standard nested stacks are deployed:**

- The admin stack will only be able to initiate remediations (through custom action or fully automated EventBridge rules) for controls which have a control runbook deployed in the target member account and Region. These are deployed by the member stack for each standard.
- The admin stack will only be able to initiate fully automated remediations using EventBridge rules for controls which have the rules deployed by the admin stack for that standard. These are deployed to the admin account.
- For simplicity, we recommend deploying standards consistently across your admin and member accounts. If you care about AWS FSBP and CIS v1.2.0, deploy those two nested admin stacks to the admin account, and deploy those two nested member stacks to each member account and Region.

**Which Control runbooks are deployed in each nested member stack:**

- The admin stack will only be able to initiate remediations (through custom action or fully automated EventBridge rules) for controls which have a control runbook deployed in the target member account and Region by the member stack for each standard.
- To exercise more fine-grained control over which controls are enabled for a particular standard, each nested stack for a standard has parameters for which control runbooks are deployed. Set the parameter for a control to the value "NOT Available" to undeploy that control runbook.

**SSM Parameters for enabling and disabling standards:**

- The admin stack will only be able to initiate remediations (through custom action or fully automated EventBridge rules) for standards that are enabled through the SSM Parameter deployed by the standard admin stack.
- To disable a standard, set the value for the SSM Parameter with the path "/Solutions/SO0111/<standard_name>/<standard_version>/status" to "No".

# Example SNS notifications

When a remediation is initiated

```
{
 "severity": "INFO",
 "message": "00000000-0000-0000-0000-000000000000: Remediation queued for SC control
 RDS.13 in account 111111111111",
 "finding": {
 "finding_id": "22222222-2222-2222-2222-222222222222",
 "finding_description": "This control checks if automatic minor version upgrades are
 enabled for the Amazon RDS database instance.",
 "standard_name": "security-control",
 "standard_version": "2.0.0",
 "standard_control": "RDS.13",
 "title": "RDS automatic minor version upgrades should be enabled",
 "region": "us-east-1",
 "account": "111111111111",
 "finding_arn": "arn:aws:securityhub:us-east-1:111111111111:security-control/RDS.13/
finding/22222222-2222-2222-2222-222222222222"
 }
}
```

### When a remediation succeeds


```
{
 "severity": "INFO",
 "message": "00000000-0000-0000-0000-000000000000: Remediation succeeded for SC
 control RDS.13 in account 111111111111: See Automation Execution output for details
 (AwsRdsDbInstance arn:aws:rds:us-east-1:111111111111:db:database-1)",
 "finding": {
 "finding_id": "22222222-2222-2222-2222-222222222222",
 "finding_description": "This control checks if automatic minor version upgrades are
 enabled for the Amazon RDS database instance.",
 "standard_name": "security-control",
 "standard_version": "2.0.0",
 "standard_control": "RDS.13",
 "title": "RDS automatic minor version upgrades should be enabled",
 "region": "us-east-1",
 "account": "111111111111",
 "finding_arn": "arn:aws:securityhub:us-east-1:111111111111:security-control/RDS.13/
finding/22222222-2222-2222-2222-222222222222"
 }
}
```


### When a remediation fails


```
{
 "severity": "ERROR",
 "message": "00000000-0000-0000-0000-000000000000: Remediation failed for SC
 control RDS.13 in account 111111111111: See Automation Execution output for details
 (AwsRdsDbInstance arn:aws:rds:us-east-1:111111111111:db:database-1)",
 "finding": {
 "finding_id": "22222222-2222-2222-2222-222222222222",
 "finding_description": "This control checks if automatic minor version upgrades are
 enabled for the Amazon RDS database instance.",
 "standard_name": "security-control",
 "standard_version": "2.0.0",
 "standard_control": "RDS.13",
 "title": "RDS automatic minor version upgrades should be enabled",
 "region": "us-east-1",
 "account": "111111111111",
 "finding_arn": "arn:aws:securityhub:us-east-1:111111111111:security-control/RDS.13/
finding/22222222-2222-2222-2222-222222222222"
 }
}
```

# Use the solution

This is a tutorial that will guide you through your first deployment of ASR. It will begin with the prerequisites for deploying the solution and it will end with you remediating example findings in a member account.

# Tutorial: Getting Started with Automated Security Response on AWS

This is a tutorial that will guide you through your first deployment. It will begin with the prerequisites for deploying the solution and it will end with you remediating example findings in a member account.

## Prepare the accounts

In order to demonstrate the cross-account and cross-Region remediation capabilities of the solution, this tutorial will use two accounts. You can also deploy the solution to a single account.

The following examples use accounts 111111111111 and 222222222222 to demonstrate the solution. 111111111111 will be the admin account and 222222222222 will be the member account. We will set up the solution to remediate findings for resources in the Regions `us-east-1` and `us-west-2`.

The table below is an example to illustrate the actions we will take for each step in each account and Region.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | None | None |
| 222222222222 | Member | None | None |

The admin account is the account that will perform the administration actions of the solution, namely initiating remediations manually or enabling fully automated remediation with EventBridge rules. This account must also be the Security Hub delegated administrator account for

all accounts in which you wish to remediate findings, but it does not need to be nor should it be the AWS Organizations administrator account for the AWS Organization to which your accounts belong.

## Enable AWS Config

Review the following documentation:

- AWS Config documentation
- AWS Config pricing
- Enabling AWS Config

Enable AWS Config in both accounts and both Regions. This will incur charges.

> **Important**
>
> Ensure that you select the option to "Include global resources (e.g., AWS IAM resources)." If you do not select this option when enabling AWS Config, you will not see findings related to global resources (e.g. AWS IAM resources)

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Enable AWS Config | Enable AWS Config |
| 222222222222 | Member | Enable AWS Config | Enable AWS Config |

## Enable AWS security hub

Review the following documentation:

- AWS Security Hub documentation
- AWS Security Hub pricing
- Enabling AWS Security Hub

Enable AWS Security Hub in both accounts and both Regions. This will incur charges.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Enable AWS Security Hub | Enable AWS Security Hub |
| 222222222222 | Member | Enable AWS Security Hub | Enable AWS Security Hub |

## Enable consolidated control findings

Review the following documentation:

- Generating and updating control findings

For the purposes of this tutorial, we will demonstrate the usage of the solution with the consolidated control findings feature of AWS Security Hub enabled, which is the recommended configuration. In partitions which do not support this feature as of the time of writing, you will need to deploy the standard-specific playbooks rather than SC (Security Control).

Enable consolidated control findings in both accounts and both Regions.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Enable consolidated control findings | Enable consolidated control findings |
| 222222222222 | Member | Enable consolidated control findings | Enable consolidated control findings |

It may take some time for findings to be generated with the new feature. You can proceed with the tutorial, but you will be unable to to remediate the findings generated without the new feature. Findings generated with the new feature can be identified by the `GeneratorId` field value `security-control/<control_id>`.

## Configure cross-Region finding aggregation

Review the following documentation:

- Cross-Region aggregation
- Enabling cross-Region aggregation

Configure finding aggregation from **us-west-2** to **us-east-1** in both accounts.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Configure aggregation from us-west-2 | None |
| 222222222222 | Member | Configure aggregation from us-west-2 | None |

It may take some time for findings to propagate to the aggregation Region. You can proceed with the tutorial, but you will be unable to remediate findings from other Regions until they begin to appear in the aggregation Region.

## Designate a Security Hub administrator account

Review the following documentation:

- Managing accounts in AWS Security Hub
- Managing organization member accounts
- Managing member accounts by invitation

In the proceeding example, we will use the manual invitation method. For a set of production accounts, we recommend managing Security Hub delegated adminstration through AWS Organizations.

From the AWS Security Hub console in the admin account (111111111111), invite the member account (222222222222) to accept the admin account as a Security Hub delegated administrator. From the member account, accept the invitation.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Invite the member account | None |
| 222222222222 | Member | Accept the invitation | None |

It may take some time for findings to propagate to the admin account. You can proceed with the tutorial, but you will be unable to remediate findings from member accounts until they begin to appear in the admin account.

## Create the roles for self-managed StackSets permissions

Review the following documentation:

- AWS CloudFormation StackSets
- Grant self-managed permissions

We will be deploying CloudFormation stacks to multiple accounts, so we will use StackSets. We cannot use service-managed permissions because the admin stack and the member stack have nested stacks, which aren't supported by the service, so we must use self-managed permissions.

Deploy the stacks for basic permissions for StackSet operations. For production accounts, you may wish to narrow the permissions according to the "advanced permissions options" documentation.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Deploy the StackSet administrator role stack<br><br>Deploy the StackSet Execution role stack | None |
| 222222222222 | Member | Deploy the StackSet execution role stack | None |

## Create the insecure resources that will generate example findings

Review the following documentation:

- Security Hub controls reference
- AWS Lambda controls

The following example resource with an insecure configuration in order to demonstrate a remediation. The example control is Lambda.1: Lambda function policies should prohibit public access.

> **Important**
>
> We will be intentionally creating a resource with an insecure configuration. Please review the nature of the control and evaluate the risk of creating such a resource in your environment for yourself. Be aware of any tooling your organization may have for detecting and reporting such resources and request an exception if appropriate. If the example control we have selected is inappropriate for you, select another control that the solution supports.

In the second Region of the member account, navigate to the AWS Lambda console and create a function in the latest Python runtime. Under Configuration → Permissions, add a policy statement to allow invoking the function from the URL with no authentication.

Confirm on the console page that the function allows public access. After the solution remediates this issue, compare the permissions to confirm that the public access has been revoked.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | None | None |
| 222222222222 | Member | None | Create a Lambda function with an insecure configura tion |

It may take some time for AWS Config to detect the insecure configuration. You can proceed with the tutorial, but you will be unable to remediate the finding until Config detects it.

## Create CloudWatch log groups for related controls

Review the following documentation:

- Monitoring CloudTrail Log Files with Amazon CloudWatch Logs
- CloudTrail controls

Various CloudTrail controls supported by the solution require there to be a CloudWatch Log group that is the destination of a multi-Region CloudTrail. In the following example, we will create a placeholder log group. For production accounts, you should properly configure CloudTrail integration with CloudWatch Logs.

Create a log group in each account and Region with the same name, for example: `asr-log-group`.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Create a log group | Create a log group |
| 222222222222 | Member | Create a log group | Create a log group |

# Deploy the solution to tutorial accounts

Gather the three Amazon S3 URLs for the admin, member, and member roles stack.

## Deploy the admin stack

View template

**automated-security-response-admin.template**

In the admin account, navigate to the CloudFormation console and deploy the admin stack into the Security Hub finding aggregation Region.

Choose No for the value of all parameters for loading nested admin stacks except for the "SC" or "Security Control" stack. This stack contains the resources for the consolidated control findings that we have configured in our accounts.

Choose No for reusing the orchestrator log group unless you have deployed this solution in this account and Region before.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Deploy the admin stack | None |
| 222222222222 | Member | None | None |

Wait until the admin stack completes deployment before continuing so a trust relationship can be created from the member accounts to the admin account.

## Deploy the member stack

View template

**automated-security-response-member.template**

In the admin account, navigate to the CloudFormation StackSets console and deploy the member stack to each account and Region. Use the StackSets admin and execution roles created in this tutorial.

Enter the name of the log group you created as the value for the parameter for the log group name.

Choose No for the value of all parameters for loading nested member stacks except for the "SC" or "security control" stack. This stack contains the resources for the consolidated control findings that we have configured in our accounts.

Enter the ID of the admin account as the value for the parameter for the admin account number. In our example, this is 111111111111.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Deploy the member StackSet / Confirm member stack deployed | Confirm member stack deployed |
| 222222222222 | Member | Confirm member stack deployed | Confirm member stack deployed |

## Deploy the member roles stack

automated-security-response-member-roles.template template button **automated-security-response-member-roles.template**

In the admin account, navigate to the CloudFormation StackSets console and deploy the member stack to each account. Use the StackSets admin and execution roles created in this tutorial. Enter the ID of the admin account as the value for the parameter for the admin account number. In our example, this is 111111111111.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Deploy the member StackSet / Confirm member stack deployed | None |
| 222222222222 | Member | Confirm member stack deployed | None |

You can proceed, but you will be unable to remediate findings until CloudFormation StackSets finishes deploying.

# Subscribe to the SNS topic

### Remediation Updates

**Topic -**{https---us-east-1-console-aws-amazon-com-sns-v3-home-region-us-east-1—topic-arn-aws-sns-us-east-1-221128147805-SO0111-ASR-Topic}[SO0111-ASR_Topic]

In the admin account, subscribe to the Amazon SNS topic created by the admin stack. This will notify you when remediations are initiated and when the succeed or fail.

**Alarms**

**Topic -**{https---us-east-1-console-aws-amazon-com-sns-v3-home-region-us-east-1—topic-arn-aws-sns-us-east-1-221128147805-SO0111-ASR-Alarm-Topic}[SO0111-ASR_Alarm_Topic]

In the admin account, subscribe to the Amazon SNS topic created by the admin stack. This will notify you when metric alarms initiate.

# Remediate example findings

In the admin account, navigate to the Security Hub console and locate the finding for the resource with an insecure configuration that you created as part of this tutorial.

This can be done in several ways:

1. In partitions which support the consolidated control findings feature, a page labeled "Controls" allows you to locate the finding by the consolidated control ID.
2. In the "Security standards" page, you can locate the control according to which standard it belongs to.
3. You can view all findings on the "Findings" page and search by attribute.

The consolidated control ID for the public Lambda Function we created is Lambda.1.

## Initiate the remediation

Select the checkbox to the left of the finding related to the resource we created. In the "Actions" drop-down menu, select "Remediate with ASR". You will see a notification that the finding was sent to Amazon EventBridge.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Initiate the remediation | None |

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 222222222222 | Member | None | None |

## Confirm that the remediation resolved the finding

You should receive two SNS notifications. The first will indicate that a remediation has been initiated, and the second will indicate that the remediation succeeded. After receiving the second notification, navigate to the Lambda console in the member account and confirm that the public access has been revoked.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | None | None |
| 222222222222 | Member | None | Confirm that the remediation succeeded |

# Trace the execution of the remediation

To understand better how the solution works, you can trace the execution of the remediation.

## EventBridge rule

In the admin account, locate an EventBridge rule named **Remediate_with_ASR_CustomAction**. This rule matches the finding you sent from Security Hub and sends it to the Orchestrator Step Functions.

## Step Functions execution

In the admin account, locate the AWS Step Functions named "**SO0111-ASR-Orchestrator**". This step function calls the SSM Automation document in the target account and Region. You can trace the execution of the remediation in the execution history of this AWS Step Functions.

### SSM Automation

In the member account, navigate to the SSM Automation console. You will find two executions of a document named "ASR-SC_2.0.0_Lambda.1" and one execution of a document named "ASR-RemoveLambdaPublicAccess".

The first execution is from the orchestrator step function in the target account. The second execution occurs in the target Region, which may not be the Region from which the finding originated. The final execution is the remediation that revokes the public access policy from the Lambda Function.

### CloudWatch Log Group

In the admin account, navigate to the CloudWatch Logs console and locate a Log Group named "**SO0111-ASR**". This log group is the destination for high-level logs from the Orchestrator Step Functions.

## Enable fully-automated remediations

The other mode of operation for the solution is to automatically remediate findings as they arrive in Security Hub.

### Confirm that you have no resources this finding may accidentally be applied to

Enabling automatic remediations will initiate remediations on all resources matching the control you enable (Lambda.1).

> **Important**
>
> Confirm that you want all public Lambda Functions within the scope of the solution to have this permission revoked. Fully-automated remediations will not be limited in scope to the Function you created. The solution will remediate this control if it is detected in any of the accounts and Regions in which it is installed.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Confirm no desired public Functions | Confirm no desired public Functions |
| 222222222222 | Member | Confirm no desired public Functions | Confirm no desired public Functions |

## Enable the rule

In the Admin account, locate an EventBridge rule named **SC_2.0.0_Lambda.1_AutoTrigger** and enable it.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Enable the automated remediation rules | None |
| 222222222222 | Member | None | None |

## Configure the resource

In the member account, re-configure the Lambda Function to allow public access.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | None | None |
| 222222222222 | Member | None | Configure the Lambda Function to allow public access |

## Confirm that the remediation resolved the finding

It may take some time for Config to detect the insecure configuration again. You should receive two SNS notifications. The first will indicate that a remediation has been initiated. The second will indicate that the remediation succeeded. After receiving the second notification, navigate to the Lambda console in the member account and confirm that the public access has been revoked.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Enable the automated remediation rules | None |
| 222222222222 | Member | None | Confirm that the remediation succeeded |

# Clean up

## Delete the example resources

In the member account, delete the example Lambda function you created.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | None | None |
| 222222222222 | Member | None | Delete the example Lambda Function |

## Delete the admin stack

In the admin account, delete the admin stack.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Delete the admin stack | None |
| 222222222222 | Member | None | None |

## Delete the member stack

In the Admin account, delete the member StackSet.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Delete the member StackSet<br><br>Confirm member stack deleted | Confirm member stack deleted |
| 222222222222 | Member | Confirm member stack deleted | Confirm member stack deleted |

## Delete the member roles stack

In the Admin account, delete the member roles StackSet.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Delete the member roles StackSet<br><br>Confirm rmember roles stack deleted | None |
| 222222222222 | Member | Confirm member roles stack deleted | None |

## Delete the retained roles

In each account, delete the retained IAM roles.

**Important**: These roles are retained for remediations which require a role in order for the remediation to continue functioning (e.g. VPC flow logging). Confirm that you do not require the continued function of any of these roles before deleting them.

Delete any roles prefixed with **SO0111-.**

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Delete retained roles | None |
| 222222222222 | Member | Delete retained roles | None |

## Schedule the retained KMS keys for deletion

The admin and member stacks both create and retain a KMS key. You will incur charges if you keep these keys.

These keys are retained in order to give you access to any resources encrypted by the solution. Confirm that you do not require them before scheduling them for deletion.

Identify the keys deployed by the solution using the aliases created by the solution or from the CloudFormation history. Schedule them for deletion.

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Identify and schedule admin key for deletion<br><br>Identify and schedule member key for deletion | Identify and schedule member key for deletion |

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 222222222222 | Member | Identify and schedule member key for deletion | Identify and schedule member key for deletion |

## Delete the stacks for self-managed StackSets permissions

Delete the stacks created to allow for self-managed StackSets permissions

| Account | Purpose | Action in us-east-1 | Action in us-west-2 |
|---|---|---|---|
| 111111111111 | Admin | Delete the StackSet administrator role stack | None |
| 222222222222 | Member | Delete the StackSet execution role stack | None |

# Developer guide

This section provides the source code for the solution and additional customizations.

## Source code

Visit our GitHub repository to download the templates and scripts for this solution, and to share your customizations with others.

## Playbooks

This solution includes the playbook remediations for the security standards defined as part of the Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0, CIS AWS Foundations Benchmark v1.4.0, CIS AWS Foundations Benchmark v3.0.0, AWS Foundational Security Best Practices (FSBP) v.1.0.0, Payment Card Industry Data Security Standard (PCI-DSS) v3.2.1, and National Institute of Standards and Technology (NIST).

If you have consolidated control findings enabled, then those controls are supported in all standards. If this feature is enabled, then only the SC playbook needs to be deployed. If not, then the playbooks are supported for the previously listed standards.

> **Important**
>
> Only deploy the playbooks for the enabled standards to avoid reaching service quotas.

For details on a specific remediation, refer to the Systems Manager automation document with the name deployed by the solution in your account. Go to the AWS Systems Manager console, then in the navigation pane choose **Documents**.

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **Total Remediations** | 63 | 34 | 29 | 33 | 65 | 19 | 90 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableAutoScalingGroupELBHealthCheck**<br><br>Auto Scaling groups associated with a load balancer should use load balancer health checks | Autoscaling.1 | | Autoscaling.1 | | Autoscaling.1 | | Autoscaling.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-ConfigureAutoScalingLaunchConfigToRequireIMDSv2**<br><br>Auto Scaling group launch configurations should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2) | | | | | Autoscaling.3 | | Autoscaling.3 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateCloudTrailMultiRegionTrail**<br><br>CloudTrail should be activated and configured with at least one multi-Region trail | CloudTrail.1 | 2.1 | CloudTrail.2 | 3.1 | CloudTrail.1 | 3.1 | CloudTrail.1 |
| **ASR-EnableEncryption**<br><br>CloudTrail should have encryption at rest activated | CloudTrail.2 | 2.7 | CloudTrail.1 | 3.7 | CloudTrail.2 | 3.5 | CloudTrail.2 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableLogFileValidation**<br><br>Ensure CloudTrail log file validation is activated | CloudTrail.4 | 2.2 | CloudTrail.3 | 3.2 | CloudTrail.4 | | CloudTrail.4 |
| **ASR-EnableCloudTrailToCloudWatchLogging**<br><br>Ensure CloudTrail trails are integrated with Amazon CloudWatch Logs | CloudTrail.5 | 2.4 | CloudTrail.4 | 3.4 | CloudTrail.5 | | CloudTrail.5 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-ConfigureS3BucketLogging**<br><br>Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket | | 2.6 | | 3.6 | | 3.4 | CloudTrail.7 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-ReplaceCodeBuildClearTextCredentials**<br><br>CodeBuild project environment variables should not contain clear text credentials | CodeBuild.2 | | CodeBuild.2 | | CodeBuild.2 | | CodeBuild.2 |
| **ASR-EnableAWSConfig**<br><br>Ensure AWS Config is activated | Config.1 | 2.5 | Config.1 | 3.5 | Config.1 | 3.3 | Config.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-MakeEBSSnapshotsPrivate**<br><br>Amazon EBS snapshots should not be publicly restorable | EC2.1 | | EC2.1 | | EC2.1 | | EC2.1 |
| **ASR-RemoveVPCDefaultSecurityGroupRules**<br><br>VPC default security group should prohibit inbound and outbound traffic | EC2.2 | 4.3 | EC2.2 | 5.3 | EC2.2 | 5.4 | EC2.2 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableVPCFlowLogs**<br><br>VPC flow logging should be enabled in all VPCs | EC2.6 | 2.9 | EC2.6 | 3.9 | EC2.6 | 3.7 | EC2.6 |
| **ASR-EnableEbsEncryptionByDefault**<br><br>EBS default encryption should be activated | EC2.7 | 2.2.1 | | | EC2.7 | 2.2.1 | EC2.7 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-RevokeUnrotatedKeys**<br>Users' access keys should be rotated every 90 days or less | IAM.3 | 1.4 | | 1.14 | IAM.3 | 1.14 | IAM.3 |
| **ASR-SetIAMPasswordPolicy**<br>IAM default password policy | IAM.7 | 1.5-1.11 | IAM.8 | 1.8 | IAM.7 | 1.8 | IAM.7 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-RevokeUnusedIAMUserCredentials**<br><br>User credentials should be turned off if not used within 90 days | IAM.8 | 1.3 | IAM.7 | | IAM.8 | | IAM.8 |
| **ASR-RevokeUnusedIAMUserCredentials**<br><br>User credentials should be turned off if not used within 45 days | | | | 1.12 | | 1.12 | IAM.22 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-RemoveLambdaPublicAccess**<br><br>Lambda functions should prohibit public access | Lambda.1 | | Lambda.1 | | Lambda.1 | | Lambda.1 |
| **ASR-MakeRDSSnapshotPrivate**<br><br>RDS snapshots should prohibit public access | RDS.1 | | RDS.1 | | RDS.1 | | RDS.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-DisablePublicAccessToRDSInstance**<br><br>RDS DB Instances should prohibit public access | RDS.2 | | RDS.2 | | RDS.2 | 2.3.3 | RDS.2 |
| **ASR-EncryptRDSSnapshot**<br><br>RDS cluster snapshots and database snapshots should be encrypted at rest | RDS.4 | | | | RDS.4 | | RDS.4 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableMultiAZOnRDSInstance**<br><br>RDS DB instances should be configured with multiple Availability Zones | RDS.5 | | | | RDS.5 | | RDS.5 |
| **ASR-EnableEnhancedMonitoringOnRDSInstance**<br><br>Enhanced monitoring should be configured for RDS DB instances and clusters | RDS.6 | | | | RDS.6 | | RDS.6 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableRDSClusterDeletionProtection**<br><br>RDS clusters should have deletion protection activated | RDS.7 | | | | RDS.7 | | RDS.7 |
| **ASR-EnableRDSInstanceDeletionProtection**<br><br>RDS DB instances should have deletion protection activated | RDS.8 | | | | RDS.8 | | RDS.8 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableMinorVersionUpgradeOnRDSDBInstance**<br><br>RDS automatic minor version upgrades should be activated | RDS.13 | | | | RDS.13 | 2.3.2 | RDS.13 |
| **ASR-EnableCopyTagsToSnapshotOnRDSCluster**<br><br>RDS DB clusters should be configured to copy tags to snapshots | RDS.16 | | | | RDS.16 | | RDS.16 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-DisablePublicAccessToRedshiftCluster**<br><br>Amazon Redshift clusters should prohibit public access | Redshift.1 | | Redshift.1 | | Redshift.1 | | Redshift.1 |
| **ASR-EnableAutomaticSnapshotsOnRedshiftCluster**<br><br>Amazon Redshift clusters should have automatic snapshots activated | Redshift.3 | | | | Redshift.3 | | Redshift.3 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableRedshiftClusterAuditLogging**<br><br>Amazon Redshift clusters should have audit logging activated | Redshift.4 | | | | Redshift.4 | | Redshift.4 |
| **ASR-EnableAutomaticVersionUpgradeOnRedshiftCluster**<br><br>Amazon Redshift should have automatic upgrades to major versions activated | Redshift.6 | | | | Redshift.6 | | Redshift.6 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-ConfigureS3PublicAccessBlock**<br><br>S3 Block Public Access setting should be activated | S3.1 | 2.3 | S3.6 | 2.1.5.1 | S3.1 | 2.1.4 | S3.1 |
| **ASR-ConfigureS3BucketPublicAccessBlock**<br><br>S3 buckets should prohibit public read access | S3.2 | | S3.2 | 2.1.5.2 | S3.2 | | S3.2 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-ConfigureS3BucketPublicAccessBlock**<br><br>S3 buckets should prohibit public write access | | S3.3 | | | | | S3.3 |
| **ASR-EnableDefaultEncryptionS3**<br><br>S3 buckets should have server-side encryption activated | S3.4 | | S3.4 | 2.1.1 | S3.4 | | S3.4 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-SetSSLBucketPolicy**<br><br>S3 buckets should require requests to use SSL | S3.5 | | S3.5 | 2.1.2 | S3.5 | 2.1.1 | S3.5 |
| **ASR-S3BlockDenylist**<br><br>Amazon S3 permissions granted to other AWS accounts in bucket policies should be restricted | S3.6 | | | | S3.6 | | S3.6 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| S3 Block Public Access setting should be activated at the bucket level | S3.8 | | | | S3.8 | | S3.8 |
| **ASR-ConfigureS3BucketPublicAccessBlock**<br><br>Ensure the S3 bucket CloudTrail logs to is not publicly accessible | | 2.3 | | | | | CloudTrail.6 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateAccessLoggingBucket**<br><br>Ensure S3 bucket access logging is activated on the CloudTrail S3 bucket | | 2.6 | | | | | CloudTrail.7 |
| **ASR-EnableKeyRotation**<br><br>Ensure rotation for customer-created CMKs is activated | | 2.8 | KMS.1 | 3.8 | KMS.4 | 3.6 | KMS.4 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for unauthorized API calls | | 3.1 | | 4.1 | | | Cloudwatch.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for AWS Management Console sign-in without MFA | | 3.2 | | 4.2 | | | Cloudwatch.2 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for usage of the "root" user | | 3.3 | CW.1 | 4.3 | | | Cloudwatch.3 |
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for IAM policy changes | | 3.4 | | 4.4 | | | Cloudwatch.4 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for CloudTrail configuration changes | | 3.5 | | 4.5 | | | Cloudwatch.5 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for AWS Management Console authentication failures | | 3.6 | | 4.6 | | | Cloudwatch.6 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs | | 3.7 | | 4.7 | | | Cloudwatch.7 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for S3 bucket policy changes | | 3.8 | | 4.8 | | | Cloudwatch.8 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for AWS Config configuration changes | | 3.9 | | 4.9 | | | Cloudwatch.9 |
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for security group changes | | 3.10 | | 4.10 | | | Cloudwatch.10 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL) | | 3.11 | | 4.11 | | | Cloudwatch.11 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for changes to network gateways | | 3.12 | | 4.12 | | | Cloudwatch.12 |
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for route table changes | | 3.13 | | 4.13 | | | Cloudwatch.13 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-CreateLogMetricFilterAndAlarm**<br><br>Ensure a log metric filter and alarm exist for VPC changes | | 3.14 | | 4.14 | | | Cloudwatch.14 |
| **AWS-DisablePublicAccessForSecurityGroup**<br><br>Ensure no security groups allow ingress from 0.0.0.0/0 to port 22 | | 4.1 | EC2.5 | | EC2.13 | | EC2.13 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **AWS-DisablePublicAccessForSecurityGroup**<br><br>Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389 | | 4.2 | | | EC2.14 | | EC2.14 |
| **ASR-ConfigureSNSTopicForStack** | CloudFormation.1 | | | | CloudFormation.1 | | CloudFormation.1 |
| **ASR-CreateIAMSupportRole** | | 1.20 | | 1.17 | | 1.17 | IAM.18 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-DisablePublicIPAutoAssign**<br><br>Amazon EC2 subnets should not automatically assign public IP addresses | EC2.15 | | | | EC2.15 | | EC2.15 |
| **ASR-EnableCloudTrailLogFileValidation** | CloudTrail.4 | 2.2 | CloudTrail.3 | 3.2 | | | CloudTrail.4 |
| **ASR-EnableEncryptionForSNSTopic** | SNS.1 | | | | SNS.1 | | SNS.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableDeliveryStatusLoggingForSNSTopic**<br><br>Logging of delivery status should be enabled for notification messages sent to a topic | SNS.2 | | | | SNS.2 | | SNS.2 |
| **ASR-EnableEncryptionForSQSQueue** | SQS.1 | | | | SQS.1 | | SQS.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-MakeRDSSnapshotPrivate** RDS snapshot should be private | RDS.1 | | RDS.1 | | | | RDS.1 |
| **ASR-BlockSSMDocumentPublicAccess** SSM Documents should not be public | SSM.4 | | | | SSM.4 | | SSM.4 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableCloudFrontDefaultRootObject**<br><br>CloudFront distributions should have a default root object configured | CloudFront.1 | | | | CloudFront.1 | | CloudFront.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-SetCloudFrontOriginDomain**<br><br>CloudFront distributions should not point to non-existent S3 origins | CloudFront.12 | | | | CloudFront.12 | | CloudFront.12 |
| **ASR-RemoveCodeBuildPrivilegedMode**<br><br>CodeBuild project environments should have a logging AWS Configuration | CodeBuild.5 | | | | CodeBuild.5 | | CodeBuild.5 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-TerminateEC2Instance**<br><br>Stopped EC2 instances should be removed after a specified time period | EC2.4 | | | | EC2.4 | | EC2.4 |
| **ASR-EnableIMDSV2OInstance**<br><br>EC2 instances should use Instance Metadata Service Version 2 (IMDSv2) | EC2.8 | | | | EC2.8 | 5.6 | EC2.8 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-RevokeUnauthorizedInboundRules**<br><br>Security groups should only allow unrestricted incoming traffic for authorized ports | EC2.18 | | | | EC2.18 | | EC2.18 |
| INSERT TITLE HERE<br><br>Security groups should not allow unrestricted access to ports with high risk | EC2.19 | | | | EC2.19 | | EC2.19 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-DisableTGWAutoAcceptSharedAttachments**<br><br>Amazon EC2 Transit Gateways should not automatically accept VPC attachment requests | EC2.23 | | | | EC2.23 | | EC2.23 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnablePrivateRepositoryScanning**<br><br>ECR private repositories should have image scanning configured | ECR.1 | | | | ECR.1 | | ECR.1 |
| **ASR-EnableGuardDuty**<br><br>GuardDuty should be enabled | GuardDuty.1 | | GuardDuty.1 | | GuardDuty.1 | | GuardDuty.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-ConfigureS3BucketLogging**<br><br>S3 bucket server access logging should be enabled | S3.9 | | | | S3.9 | | S3.9 |
| **ASR-EnableBucketEventNotifications**<br><br>S3 buckets should have event notifications enabled | S3.11 | | | | S3.11 | | S3.11 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-SetS3LifecyclePolicy**<br>S3 buckets should have lifecycle policies configured | S3.13 | | | | S3.13 | | S3.13 |
| **ASR-EnableAutoSecretRotation**<br>Secrets Manager secrets should have automatic rotation enabled | SecretsManager.1 | | | | SecretsManager.1 | | SecretsManager.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-RemoveUnusedSecret**<br><br>Remove unused Secrets Manager secrets | SecretsManager.3 | | | | SecretsManager.3 | | SecretsManager.3 |
| **ASR-UpdateSecretRotationPeriod**<br><br>Secrets Manager secrets should be rotated within a specified number of days | SecretsManager.4 | | | | SecretsManager.4 | | SecretsManager.4 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableAPIGatewayCacheDataEncryption**<br><br>API Gateway REST API cache data should be encrypted at rest | | | | | APIGateway.5 | | APIGateway.5 |
| **ASR-SetLogGroupRetentionDays**<br><br>CloudWatch log groups should be retained for a specified time period | | | | | CloudWatch.16 | | CloudWatch.16 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-AttachServiceVPCEndpoint**<br><br>Amazon EC2 should be configured to use VPC endpoints that are created for the Amazon EC2 service | EC2.10 | | | | EC2.10 | | EC2.10 |
| **ASR-TagGuardDutyResource**<br><br>GuardDuty filters should be tagged | | | | | | | GuardDuty.2 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-TagGuardDutyResource**<br><br>GuardDuty detectors should be tagged | | | | | | | GuardDuty.4 |
| **ASR-AttachSSMPermissionsToEC2**<br><br>Amazon EC2 instances should be managed by Systems Manager | SSM.1 | | SSM.3 | | | | SSM.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-ConfigureLaunchConfigNoPublicIPDocument**<br><br>Amazon EC2 instances launched using Auto Scaling group launch configurations should not have public IP addresses | | | | | Autoscaling.5 | | Autoscaling.5 |
| **ASR-EnableAPIGatewayExecutionLogs** | APIGateway.1 | | | | | | APIGateway.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableMacie**<br><br>Amazon Macie should be enabled | Macie.1 | | | | Macie.1 | | Macie.1 |
| **ASR-EnableAthenaWorkGroupLogging**<br><br>Athena workgroups should have logging enabled | Athena.4 | | | | | | Athena.4 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnforceHTTPSForALB**<br><br>Application Load Balancer should be configured to redirect all HTTP requests to HTTPS | ELB.1 | | ELB.1 | | ELB.1 | | ELB.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-LimitECSRootFilesystemAccess**<br><br>ECS containers should be limited to read-only access to root filesystems | ECS.5 | | | | ECS.5 | | ECS.5 |
| **ASR-EnableElastiCacheBackups**<br><br>ElastiCache (Redis OSS) clusters should have automatic backups enabled | ElastiCache.1 | | | | ElastiCache.1 | | ElastiCache.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableElastiCacheVersionUpgrades**<br><br>ElastiCache clusters should have automatic minor version upgrades enabled | ElastiCache.2 | | | | ElastiCache.2 | | ElastiCache.2 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-EnableElastiCacheReplicationGroupFailover**<br><br>ElastiCache replication groups should have automatic failover enabled | ElastiCache.3 | | | | ElastiCache.3 | | ElastiCache.3 |
| **ASR-ConfigureDynamoDBAutoScaling**<br><br>DynamoDB tables should automatically scale capacity with demand | DynamoDB.1 | | | | DynamoDB.1 | | DynamoDB.1 |

| Description | AWS FSBP | CIS v1.2.0 | PCI v3.2.1 | CIS v1.4.0 | NIST | CIS v3.0.0 | Security control ID |
|---|---|---|---|---|---|---|---|
| **ASR-TagDynamoDBTableResource**<br><br>DynamoDB tables should be tagged | | | | | | | DynamoDB.5 |
| **ASR-EnableDynamoDDeletionProtection**<br><br>DynamoDB tables should have deletion protection enabled | | | | | DynamoDB.6 | | DynamoDB.6 |

# Adding new remediations

Remediations can be added manually by updating the appropriate playbook files, or programmatically by extending the solution through CDK constructs, depending on your preferred workflow.

> **Note**
>
> The instructions that follow leverage resources installed by the solution as a starting point. By convention, most solution resource names contain **ASR** and/or **SO0111** to make it easy to locate and identify them.

## Overview of manually workflow

Automated Security Response on AWS runbooks must follow the following standard naming:

ASR-*<standard>*-*<version>*-*<control>*

**Standard**: The abbreviation for the security standard. This must match standards supported by ASR. It must be one of "CIS", "AFSBP", "PCI", "NIST", or "SC".

**Version**: The version of the standard. Again, this must match the version supported by ASR and the version in the finding data.

**Control**: The control ID of the control to be remediated. This must match the finding data.

1. Create a runbook in the member account(s).
2. Create an IAM role in the member account(s).
3. (Optional) Create an automatic remediation rule in the admin account.

### Step 1. Create a runbook in the member account(s)

1. Sign in to the AWS Systems Manager console and obtain an example of the finding JSON.
2. Create an automation runbook that remediates the finding. In the **Owned by me** tab, use any of the ASR- documents under the **Documents** tab as a starting point.
3. The AWS Step Functions in the admin account will run your runbook. Your runbook must specify the remediation role in order to be passed when calling the runbook.

### Step 2. Create an IAM role in the member account(s)

1. Sign in to the AWS Identity and Access Management console.

2. Obtain an example from the IAM **SO0111** roles and create a new role. The role name must start with SO0111-Remediate-*<standard>*-*<version>*-*<control>*. For example, if adding CIS v1.2.0 control 5.6 the role must be `SO0111-Remediate-CIS-1.2.0-5.6`.
3. Using the example, create a properly scoped role that allows only the necessary API calls to perform remediation.

At this point, your remediation is active and available for automated remediation from the ASR Custom Action in AWS Security Hub.

### Step 3: (Optional) Create an automatic remediation rule in the admin account

Automatic (not "automated") remediation is the immediate execution of the remediation as soon as the finding is received by AWS Security Hub. Carefully consider the risks before using this option.

1. View an example rule for the same security standard in CloudWatch Events. The naming standard for rules is `standard_control_*AutoTrigger*`.
2. Copy the event pattern from the example to be used.
3. Change the `GeneratorId` value to match the `GeneratorId` in your Finding JSON.
4. Save and activate the rule.

## Overview of CDK workflow

In summary, the following files in the ASR repo will be modified or added. In this example, a new remediation for ElastiCache.2 was added to the SC and AFSBP playbooks.

> **Note**
>
> All new remediations should be added to the SC playbook, since it consolidates all remediations available in ASR. If you intend to deploy only a specific set of playbooks (e.g., AFSBP), then you can either: (1) add the remediation to **only** your intended playbook(s), or (2) add the remediation to all playbooks for which it exists in the corresponding Security Hub Standard, in addition to the SC playbook. The second option is recommended for flexibility.

In this example, ElastiCache.2 is included in the following Security Hub Standards:

- AFSBP
- NIST.800-53.r5 SI-2
- NIST.800-53.r5 SI-2(2)
- NIST.800-53.r5 SI-2(4)
- NIST.800-53.r5 SI-2(5)
- PCI DSS v4.0.1/6.3.3

Since, by default, ASR only implements playbooks for AFSBP and NIST.800-53, we will add this new remediation to those playbooks in addition to SC.

**Modify**

- source/lib/remediation-runbook-stack.ts
- source/playbooks/AFSBP/lib/[standard name]_remediations.ts
- source/playbooks/NIST80053/lib/control_runbooks-construct.ts
- source/playbooks/NIST80053/lib/[standard name]_remediations.ts
- source/playbooks/SC/lib/control_runbooks-construct.ts
- source/playbooks/SC/lib/sc_remediations.ts
- source/test/regex_registry.ts

**Add**

- source/playbooks/SC/ssmdocs/SC_ElastiCache.2.ts
- source/playbooks/SC/ssmdocs/descriptions/ElastiCache.2.md
- source/remediation_runbooks/EnableElastiCacheVersionUpgrades.yaml

> **Note**
>
> The name chosen for the runbook can be any string, as long as it is consistent with the rest of the changes made.

- source/playbooks/NIST80053/ssmdocs/NIST80053_ElastiCache.2.ts
- source/playbooks/AFSBP/ssmdocs/AFSBP_ElastiCache.2.yaml

**Development steps**

1. Create the Remediation Runbook.
2. Create the Control Runbooks.
3. Integrate Each Control Runbook with a Playbook.
4. Create the Remediation IAM Role & Integrate Remediation Runbook
5. Update Unit Tests

## Step 1: Create the Remediation Runbook

This is the SSM document used to remediate resources. It must include the `AutomationAssumeRole` parameter, which is the IAM role with permissions to execute the remediation. View the existing file `source/remediation_runbooks/EnableElastiCacheVersionUpgrades.yaml` as a reference when creating new remediation runbooks.

All new runbooks should be added to the `source/remediation_runbooks/` directory.

## Step 2: Create the Control Runbooks

A control runbook is a playbook-specific runbook that parses the finding data from the given standard and executes the appropriate Remediation Runbook. Since we are adding the ElastiCache.2 remediation to the SC, AFSBP, and NIST80053 playbooks, we must create a new control runbook for each. The following files are created:

- source/playbooks/SC/ssmdocs/SC_ElastiCache.2.ts
- source/playbooks/NIST80053/ssmdocs/NIST80053_ElastiCache.2.ts
- source/playbooks/AFSBP/ssmdocs/AFSBP_ElastiCache.2.yaml

**Example**

The naming of these files is important and must follow the format <PLAYBOOK_NAME>_<CONTROL.ID>.ts/yaml

Some playbooks in ASR support IaC control runbooks in TypeScript, while others must be written in raw YAML. Reference the existing remediations in the respective playbook as examples. In this example, we will cover the SC playbook, which uses IaC.

In the SC playbook, your new control runbook should export a class that extends ControlRunbookDocument and matches the name of your remediation runbook. Take a look at the example below:

```
export class EnableElastiCacheVersionUpgrades extends ControlRunbookDocument {
  constructor(scope: Construct, id: string, props: ControlRunbookProps) {
    super(scope, id, {
      ...props,
      securityControlId: 'ElastiCache.2',
      remediationName: 'EnableElastiCacheVersionUpgrades',
      scope: RemediationScope.REGIONAL,
      resourceIdRegex: <Regex>,
      resourceIdName: 'ClusterId',
      updateDescription: new StringFormat('Automatic minor version upgrades enabled for
 cluster %s.', [
        StringVariable.of(`ParseInput.ClusterId`),
      ]),
    });
  }
}
```

- `securityControlId` is the control ID for the remediation that you are adding, as it is defined in the consolidated controls view in Security Hub.
- `remediationName` is the name you have chosen for your remediation runbook.
- `scope` is the scope of the resource you are remediating, indicating whether it exists globally or in a specific region.
- `resourceIdRegex` is the regex used to capture the resource ID that you would like to pass to the remediation runbook as a parameter. Only one group should be captured, all other groups should be non-capturing. If you would like to pass the entire ARN, omit this field.
- `resourceIdName` is the name you would like to set for the resource ID captured using `resourceIdRegex`, this should match the resource ID parameter name in your remediation runbook.
- `updateDescription` is the string you would like to assign to the "notes" section of the finding in Security Hub once the remediation succeeds.

You must also export a function called `createControlRunbook` which returns a new instance of your class. For ElastiCache.2, this looks like:

```
export function createControlRunbook(scope: Construct, id: string, props:
 PlaybookProps): ControlRunbookDocument {
  return new EnableElastiCacheVersionUpgrades(scope, id, { ...props, controlId:
 'ElastiCache.2' });
}
```

where `controlId` is the control ID as defined in the Security Standard associated with the playbook under which you are operating.

If the Security Hub control has parameters that you would like to pass to your remediation runbook, you can pass them by adding overrides to the following methods: - `getExtraSteps`: defines default values for each parameter implemented for the control in Security Hub

> **Note**
>
> Each parameter from Security Hub must be given a default value

- `getInputParamsStepOutput`: defines the outputs for the GetInputParams step of the control runbook
- Each output has a `name`, `outputType`, and `selector`. The `selector` should be the same selector used in the `getExtraSteps` method override.
- `getRemediationParams`: defines the parameters passed to the remediation runbook, fetched from the GetInputParams step outputs.

To view an example, navigate to the `source/playbooks/SC/ssmdocs/SC_DynamoDB.1.ts` file.

### Step 3: Integrate Each Control Runbook with a Playbook

For each control runbook created in the previous step, you must now integrate it with the infrastructure definitions in the associated playbook. Follow the steps below for each control runbook.

> **Important**
>
> If you created the control runbook using raw YAML instead of typescript IaC, skip to the next section.

In */<playbook_name>/*`control_runbooks-construct.ts` Import your newly created control runbook file like:

```
import * as elasticache_2 from '../ssmdocs/SC_ElastiCache.2';
```

Next, go to the array for

```
const controlRunbooksRecord: Record<string, any>
```

And add a new entry mapping the control ID (playbook-specific) to the `createControlRunbook` method you've created:

```
'ElastiCache.2': elasticache_2.createControlRunbook,
```

Add the playbook-specific control ID to the list of remediations in *<playbook_name>*`\_remediations.ts` like below:

```
{ control: 'ElastiCache.2', versionAdded: '2.3.0' },
```

The `versionAdded` field should be the latest version of the solution. If adding the remediation breaches the template size limit, increase the `versionAdded`. You can adjust the number of remediations included in each playbook member stack in `solution_env.sh`.

## Step 4: Create the Remediation IAM Role & Integrate Remediation Runbook

Each remediation has its own IAM role with custom permissions required to execute the remediation runbook. In addition, the `RunbookFactory.createRemediationRunbook` method needs to be invoked to add the remediation runbook you created in Step 1 to the solution's CloudFormation templates.

In the `remediation-runook-stack.ts`, each remediation has its own code block in the `RemediationRunbookStack` class. The following code block shows the creation of a new IAM role and remediation runbook integration for the ElastiCache.2 remediation:

```
    //-----------------------
    // EnableElastiCacheVersionUpgrades
    //
    {
      const remediationName = 'EnableElastiCacheVersionUpgrades'; // should match the
 name of your remediation runbook
```

```
      const inlinePolicy = new Policy(props.roleStack, `ASR-Remediation-Policy-
${remediationName}`);

      const remediationPolicy = new PolicyStatement();
      remediationPolicy.addActions('elasticache:ModifyCacheCluster');
      remediationPolicy.effect = Effect.ALLOW;
      remediationPolicy.addResources(`arn:${this.partition}:elasticache:*:
${this.account}:cluster:*`);
      inlinePolicy.addStatements(remediationPolicy);

      new SsmRole(props.roleStack, 'RemediationRole ' + remediationName, { // creates
 the remediation IAM role
        solutionId: props.solutionId,
        ssmDocName: remediationName,
        remediationPolicy: inlinePolicy,
        remediationRoleName: `${remediationRoleNameBase}${remediationName}`,
      });

      RunbookFactory.createRemediationRunbook(this, 'ASR ' + remediationName, { // adds
 the remediation runbook to the solution's cloudformation templates
        ssmDocName: remediationName,
        ssmDocPath: ssmdocs,
        ssmDocFileName: `${remediationName}.yaml`,
        scriptPath: `${ssmdocs}/scripts`,
        solutionVersion: props.solutionVersion,
        solutionDistBucket: props.solutionDistBucket,
        solutionId: props.solutionId,
        namespace: namespace,
      });
    }
```

## Step 5: Update Unit Tests

We recommend updating and running the unit tests after adding a new remediation.

First, you must add any new regular expressions (that are not already added) into the `source/test/regex_registry.ts` file. This file enforces testing for each new regular expression included in the solution's runbooks. Take a look at the `addElastiCacheClusterTestCases` function as an example, which is used to test regular expressions used in ElastiCache remediations.

Finally, you'll need to update the snapshots for each stack. Snapshots are version-controlled CloudFormation template definitions that are used to track changes made to ASR's infrastructure.

You can update these snapshot files by running the following command from the `deployment` directory:

```
./run-unit-tests.sh update
```

Now you are ready to deploy your new remediation! Navigate to the **Build and Deploy** section below for instructions on building and deploying the solution with your new changes.

## Adding a new playbook

Download the Automated Security Response on AWS solution playbooks and deployment source code from the GitHub repository.

The AWS CloudFormation resources are created from AWS CDK components, and the resources contain the playbook template code that you can use to create and configure new playbooks. For more information about setting up your project and customizing your playbooks, refer to the README.md file in GitHub.

## AWS Systems Manager Parameter Store

Automated Security Response on AWS uses AWS Systems Manager Parameter Store for storage of operational data. The following parameters are stored in Parameter Store:

| Name | Value | Use |
|---|---|---|
| `/Solutions/SO0111/CMK_REMEDIATION_ARN` | AWS KMS key that will encrypt data for FSBP remediations | Encryption of customer data, such as CloudTrail logs, as part of remediations |
| `/Solutions/SO0111/CMK_ARN` | AWS KMS key that ASR will use to encrypt data | Encryption of solution data |
| `/Solutions/SO0111/SNS_Topic_ARN` | ARN of the Amazon SNS topic for the solution | Notification of remediation events |
| `/Solutions/SO0111/SNS_Topic_Config.1` | SNS topic for AWS Config updates | Config.1 remediation |

| Name | Value | Use |
| --- | --- | --- |
| /Solutions/SO0111/sendAnonymousMetrics | Yes | Anonymized metrics collection |
| /Solutions/SO0111/version | Solution version | |
| /Solutions/SO0111/*<security standard long name>*/*<version>* /status | enabled | Indicates whether the standard is active in the solution. A standard can be disabled for automated remediation by changing this to disabled |
| /Solutions/SO0111/*<security standard long name>*/shortname | String | Short name for the security standard. For example: CIS, AFSBP, PCI |
| /Solutions/SO0111/*<security standard long name>*/*<version>* /*<control>* /remap | String | When one control uses the same remediation as another, these parameters accomplish the remap |

## Amazon SNS topic - Remediation Progress

Automated Security Response on AWS creates an Amazon SNS topic, SO0111-ASR_Topic. This topic is used to post updates about remediation progress. Following are the three possible notifications sent to this topic.

```
Remediation queued for [.replaceable]`<standard>` control [.replaceable]`<control_ID>`
 in account [.replaceable]`<account_ID>`
```

```
Remediation failed for [.replaceable]`<standard>` control [.replaceable]`<control_ID>`
 in account [.replaceable]`<account_ID>`
```

```
[.replaceable]`<control_ID>` remediation was successfully invoke via AWS Systems
 Manager in account [.replaceable]`<account_ID>`
```

This is the completion message. It indicates that the remediation completed without error; however, the definitive test for successful remediation is the AWS Config check and/or manual validation.

## Filtering an SNS topic subscription

Amazon SNS subscription filter policies:

1. Navigate to the subscription of the SNS topic.
2. Under Subscription filter policy, select"Edit".
3. Expand "Subscription filter policy" and toggle the "Subscription filter policy" option to enable filters.
4. Select the "Message Body" scope.
5. Add your policy to the JSON editor.
6. Save changes.

Example policies:

Filter by account

```
{
"finding": {
"account": [
"111111111111",
"222222222222"
]
}
```

Filter for errors

```
{
```

```
"severity": ["ERROR"]
}
```

Filter by controls

```
{
"finding": {
"standard_control": ["S3.9","S3.6"]
}
}
```

## Amazon SNS topic - CloudWatch Alarms

This solution creates an Amazon SNS topic, `SO0111-ASR_Alarm_Topic`. This topic is used to post alarm alerts.

Details of any Alarms that enter the ALARM state will be sent to this topic.

## Initiate Runbook on Config Findings

This solution can initiate runbooks based on custom AWS Config findings. To do this you will need to:

1. Find the AWS Config rule name that you would like to remediate. This can be found in either AWS Config or in the finding that Security Hub generates for this rule.
2. Navigate to AWS Systems Manager Parameter Store and select Create Parameter.
3. The name of your rule should be `/Solutions/SO0111/`[.replaceable]`Rule name from Step 1`
4. The value should be formatted as such:

```
{

"RunbookName":"Name of SSM runbook",

"RunbookRole": "Role that Orchestrator will assume"

}
```

1. RunbookName is a required field and will be the runbook that is run when you remediate this Config rule. RunbookRole is the role that the orchestrator will assume when running this role. It is not a required field, and if left out, the orchestrator will default to using the account's member role.
2. Once this is in place, you can remediate your Config rule using the "Remediate with ASR" custom action found on the Security Hub.

# Reference

This section includes information about an optional feature for collecting unique metrics for this solution, pointers to related resources, and a list of builders who contributed to this solution.

## Anonymized data collection

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When enabled, the following information is collected and sent to AWS:

- **Solution ID** - The AWS solution identifier
- **Unique ID (UUID)** - Randomly generated, unique identifier for each AWS Security Hub Response and Remediation deployment
- **Timestamp** - Data collection timestamp
- **Instance Data** - Information about this stack deployment
- **Solution configuration** - Features turned on and parameters set during initial launch
- **Status** - Deployment status (passed or failed solution) or (passed or failed remediation)
- **Error message** - The generic error message in the status field
- **Generator_id** - Security Hub rule information
- **Type** - Remediation type and name
- **productArn** - The Region where Security Hub is deployed
- **finding_triggered_by** - The type of remediation performed (custom action or automated trigger)

AWS owns the data gathered through this survey. Data collection is subject to the AWS Privacy Notice. To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the AWS CloudFormation template to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

```
Mappings:
Solution:
```

```
Data:
SendAnonymizedUsageData: 'Yes'
```

to:

```
Mappings:
Solution:
Data:
SendAnonymizedUsageData: 'No'
```

4. Sign in to the AWS CloudFormation console.
5. Select **Create stack**.
6. On the **Create stack** page, **Specify template section**, select **Upload a template file**.
7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in Launch the stack in the Automated deployment section of this guide.

## Related resources

- Automated Response and Remediation with AWS Security Hub
- CIS Amazon Web Services Foundations benchmarks, version 1.2.0
- AWS Foundational Security Best Practices standard
- Payment Card Industry Data Security Standard (PCI DSS)
- National Institute of Standards and Technology (NIST) SP 800-53 Rev. 5

## Contributors

The following individuals contributed to this document:

- Mike O'Brien
- Nikhil Reddy
- Chandini Penmetsa
- Chaitanya Deolankar
- Max Granat

- Tim Mekari
- Aaron Schuetter
- Andrew Yankowsky
- Josh Moss
- Ryan Garay
- Thiemo Belmega
- Mykhailo Markhain

# Revisions

Publication date: *August 2020 (last update: January 2025)*

Visit the CHANGELOG.md in our GitHub repository to track version-specific improvements and fixes.

# Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. AWS responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Automated Security Response on AWS is licensed under the terms of the of the Apache License Version 2.0 available at The Apache Software Foundation.