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. Authorized users can start remediations from the **Actions** menu in the AWS Security Hub Cloud Security Posture Management (CSPM) console, from the solution's Web UI, or programmatically through the solution's API, and can 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 Remediation Configuration DynamoDB table. To revert these changes, you must manually put resources back in their original state.
>
> When remediating AWS resources deployed as 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) v1.0.0

- 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. We recommend using the SC playbook along with consolidated control findings in Security Hub.

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

Configure the solution to automatically remediate findings for specific controls by modifying the Remediation Configuration DynamoDB table deployed to the admin account.

Remediate findings from multiple AWS security services

Beyond AWS Security Hub CSPM control findings, the solution can remediate findings from Amazon Inspector, Amazon GuardDuty, and Amazon Macie. The pre-processor maps each finding to the appropriate remediation and processes both the AWS Security Finding Format (ASFF) and the Open Cybersecurity Schema Framework (OCSF).

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

The solution publishes to an Amazon SNS topic when remediations are initiated and when they succeed or fail. You can also create notification configurations from the Web UI to deliver findings and remediation results to email, Slack, JIRA, or ServiceNow. For more information, see the Configure notifications section.

Use the Web User Interface to start, view, and manage remediations

You have the option to enable the solution's Web UI when deploying the Admin stack. The Web UI provides a comprehensive, user-friendly view to run remediations and view all past remediations performed by the solution.

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.

To accelerate building these customizations, the solution provides the AI Toolkit for Custom Remediations. The toolkit delivers ASR best practices, guardrails, and development patterns as an instruction prompt that you provide to an AI assistant in your integrated development environment (IDE), giving the assistant the context it needs to help you author production-ready custom remediations more efficiently and safely.

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.

[Solution Web UI] delegated admin

In the context of the solution's Web UI, a delegated admin is a user that has been invited by the admin and has full access to run remediations and view remediation history. This user can also view and manage other Account Operator users.

[Solution Web UI] account operator

In the context of the solution's Web UI, an account operator is a user invited by an admin or delegated admin to access the solution's Web UI. This user is associated with a list of AWS Account Ids provided in their invitation; they may only run remediations and view remediation history as it pertains to resources in these accounts.

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

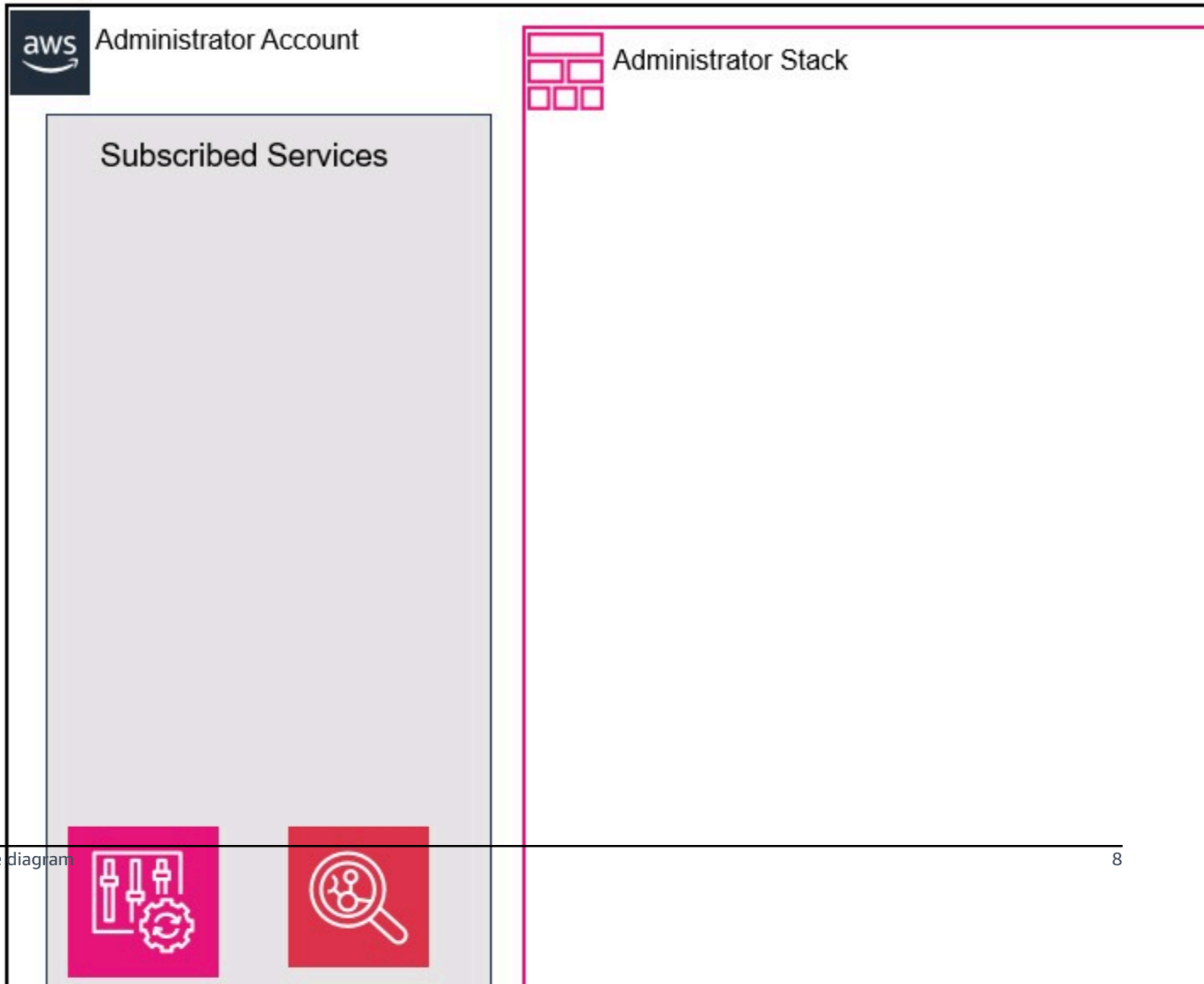
Architecture overview

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


Administrator Account
Administrator Stack
Subscribed Services

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

The high-level 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 GuardDuty, 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. **Listen**: EventBridge events are emitted by AWS Security Hub for every finding created or modified by the service. Automated Security Response on AWS (ASR) deploys two EventBridge rules that listen for finding events generated by AWS Security Hub:
   - **Custom Action** EventBridge Rule: Listens for custom actions events emitted by AWS Security Hub CSPM when the 'Remediate with ASR' custom action is triggered by a user. The event is forwarded to the Orchestrator for remediation.
   - **Findings** EventBridge Rule: Listens for all finding create or update events emitted by AWS Security Hub and AWS Security Hub CSPM. The rule forwards these events to the Pre-Processor's SQS queue for further processing. The Pre-Processor maps each finding to the appropriate remediation. In addition to AWS Security Hub CSPM control findings, the solution can remediate findings from additional AWS security services — Amazon Inspector, Amazon GuardDuty, and Amazon Macie. The solution processes both the AWS Security Finding Format (ASFF) used by AWS Security Hub CSPM and the Open Cybersecurity Schema Framework (OCSF) used by AWS Security Hub v2.
3. **Initiate**: You can initiate remediations by hand, or configure them to run automatically. To run a remediation manually, you can use the Web UI deployed by the solution or the custom actions feature in AWS Security Hub CSPM. 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. To configure remediations to run automatically, see the Enable fully-automated remediations page.

4. **Pre-remediate**: In the admin account, AWS Step Functions processes the remediation event and prepares it to be scheduled.
5. **Schedule**: The solution invokes the scheduling AWS Lambda function to place the remediation event in the Amazon DynamoDB state table.
6. **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.
7. **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.
8. **(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.
9. **Notify and log**: The playbook logs the results to a CloudWatch log group, updates the AWS Security Hub finding, and maintains an audit trail of actions in the finding notes. The solution then distributes and monitors the outcome through a dedicated notification pipeline:
   a. **Queue and dispatch the notification**: The Orchestrator sends the remediation result to a notification Amazon Simple Queue Service (Amazon SQS) queue. A notification dispatcher Lambda function consumes the queue and fans the result out through a solution-managed Amazon Simple Notification Service (Amazon SNS) topic. In addition to the solution-managed SNS topic, you can create notification configurations from the Web UI to deliver findings and remediation results to email, Slack, JIRA, ServiceNow, or an SNS topic. For more information, see the Configure notifications section.
   b. **Monitor delivery**: The solution's CloudWatch alarms publish to an SNS alarm topic, and the ASR Amazon CloudWatch dashboard surfaces remediation and notification-pipeline health so you can detect delivery failures.

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 solution requires wildcard permissions to be able to act on any resources.
- For security purposes, all IAM roles deployed by this solution use least-privilege permissions scoped to the specific resources they manage.

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 CSPM's custom action feature. When AWS Security Hub CSPM console users choose **Actions > Remediate with ASR**, the selected findings are sent to EventBridge and trigger the remediation workflow.

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 configure fully-automated remediations on a per-control basis using Amazon DynamoDB. 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 after installation by modifying the Remediation Configuration DynamoDB table.

Cross-account remediation

Automated Security Response on AWS uses cross-account roles to work across primary and secondary accounts. 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 Playbooks (in the Developer Guide). 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.

After deploying the solution's CloudFormation stacks, the playbooks are ready to use immediately—no additional configuration is required to enable remediations for the Security Standards listed above.

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.

Multi-service remediation

In addition to AWS Security Hub CSPM control findings (identified by control IDs such as `S3.1` or `IAM.7`), the solution can remediate findings from additional AWS security services. These findings are categorized by service name, finding type, and resource type rather than by a

Security Hub control ID. The preprocessor maps each service and finding type to the appropriate remediation. Multi-service remediations use the same two-tier runbook architecture, ingestion, and orchestration flow as control-based remediations. Configure them the same way through the Remediation Configuration Amazon DynamoDB table.

This release supports one finding type per service:

Service	Control ID	Supported resource types	Remediation	Manual follow-up required
Amazon Inspector	`Inspector.InstanceVulnerability`	EC2 instances managed by AWS Systems Manager	Runs the `AWS-RunPatchBaseline` Systems Manager command to install the specific vulnerable packages identified in the finding. The Systems Manager Agent must be version 2.0.834.0 or later.	No
Amazon GuardDuty	`GuardDuty.IAMUser`	IAM access keys	Contains the potentially compromised IAM user by disabling its access keys, removing console access, and attaching a deny-all policy, using the AWS-managed `AWSSupport-ContainIAMPrincipal` runbook. The solution backs up the original configuration so you can roll back the containment from the Web UI.	Yes
Amazon Macie	`Macie.SensitiveDataS3Object`	S3 buckets containing objects with sensitive data	Enables S3 Block Public Access on the bucket to prevent public exposure of the sensitive data.	Yes

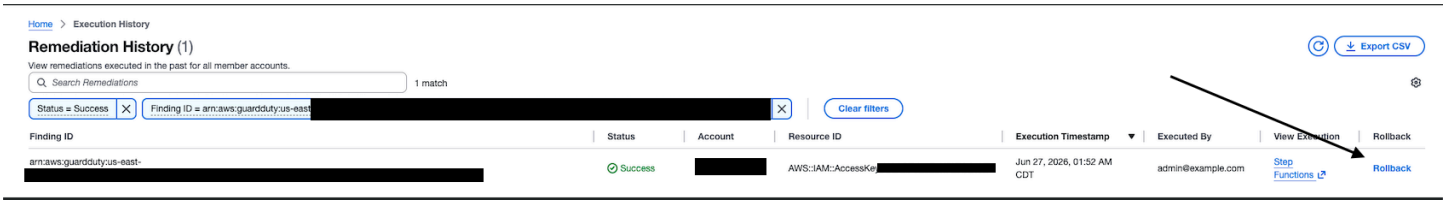
Manual follow-up required

The Amazon GuardDuty and Amazon Macie remediations are a first line of defense and do not resolve the underlying finding. They contain or limit exposure of the affected resource, but a security team must investigate and complete service-specific follow-up before resolving the finding. The finding remains visible in the solution for that follow-up.
For `GuardDuty.IAMUser`, the containment can be rolled back from the Web UI. If auto-remediation is enabled for the control, the Web UI warns you when you start a rollback that the solution might re-remediate the finding, because rolling back leaves the finding eligible for the solution to act on again. Disable auto-remediation for the control before rolling back if you want the rollback to persist.
Because a rollback restores the resource to its pre-remediation (non-compliant) state, the underlying finding remains unresolved. Re-enabling auto-remediation for the control while that finding is still unresolved makes it eligible again, so the solution re-applies the remediation the next time AWS Security Hub emits an event for the finding. Keep auto-remediation disabled for the control until you have investigated and resolved the underlying finding.

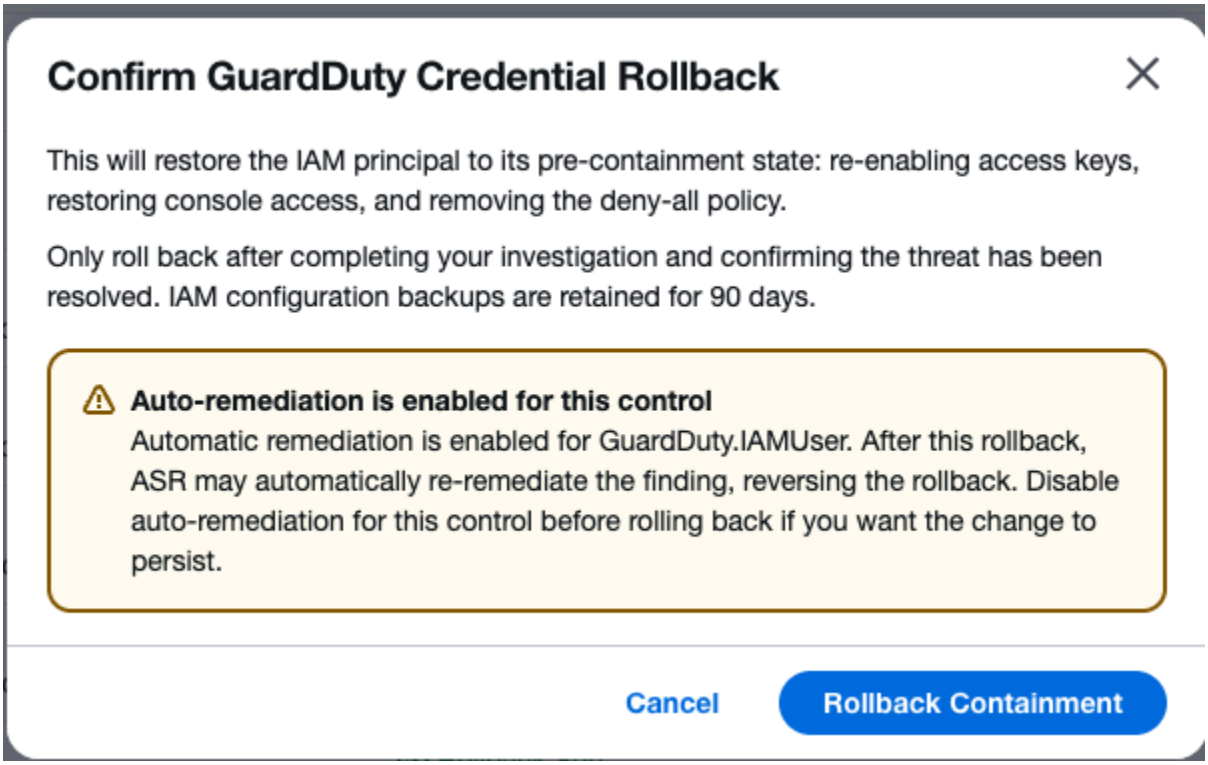
GuardDuty.IAMUser temporary credentials

The `AWSSupport-ContainIAMPrincipal` runbook contains a long-term IAM user (its access key ID begins with `AKIA`). A `GuardDuty.IAMUser` finding whose compromised principal is a temporary credential from an assumed-role or federated session — its access key ID begins with `ASIA` — has no IAM user to contain, so every remediation attempt would fail. The solution does not automatically remediate these findings; it records them for visibility but never triggers the remediation. Investigate the originating role or identity provider and revoke the active session directly.

GuardDuty.IAMUser rollback button in the Web UI Remediation History



GuardDuty.IAMUser rollback warning in the Web UI



Finding formats

Security Hub represents findings in two schema formats, and the solution processes both:

- **ASFF (AWS Security Finding Format)** – The format used by AWS Security Hub CSPM (Classic). The solution processes AWS Security Hub CSPM control findings in ASFF.
- **OCSF (Open Cybersecurity Schema Framework)** – The format used by AWS Security Hub v2. The solution processes Amazon Inspector, Amazon GuardDuty, and Amazon Macie findings in their native OCSF format.

Rather than forcing every finding into a single schema, the solution processes each finding in the format its source service emits. Existing remediations that use AWS Config continue to work unchanged in ASFF, while multi-service findings are processed in their native OCSF format.

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**. EventBridge rules are used to listen and trigger on events emitted by AWS Security Hub and AWS Security Hub CSPM.
AWS IAM	**Core**. Deploys many roles to allow remediati ons on different resources.
AWS Lambda	**Core.** Deploys multiple Lambda functions used by the Step Functions orchestrator to remediate issues. Serves as the backend for the solution's Web UI integrated with API Gateway.
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 Automation Documents that contain the remediation logic to be executed by the solution.

AWS service	Description
	Uses Parameter Store to maintain solution metadata and configuration settings.
Amazon DynamoDB	**Core**. Stores the last run remediation in each account and Region to optimize scheduling of remediations. Stores findings generated by AWS Security Hub & AWS Security Hub CSPM. Stores remediation and solution configuration metadata. Stores data for users accessing the solution's Web UI.
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. Collects metrics to display on a custom dashboard with alarms.
Amazon Simple Notification Service	**Supporting**. Deploys SNS topics that receive a notification once a remediation has been completed.
Amazon SQS	**Supporting**. Assists with scheduling remediati ons so that the solution can run remediations in parallel. Buffers Lambda executions using Lambda EventSource Mappings.
AWS Key Management Service	**Supporting**. Used to encrypt data for remediations.

AWS service	Description
AWS Config	**Supporting**. Records all resources for use with AWS Security Hub.
Amazon Inspector	**Supporting**. The solution ingests Amazon Inspector `Inspector.Instance Vulnerability` findings and remediates the affected Amazon EC2 instances.
Amazon GuardDuty	**Supporting**. The solution ingests Amazon GuardDuty `GuardDuty.IAMUser` findings and contains the affected IAM principal.
Amazon Macie	**Supporting**. The solution ingests Amazon Macie `Macie.SensitiveDataS3Object` findings and blocks public access to the affected Amazon S3 bucket.
Amazon S3	**Supporting**. Stores exported remediation history and log data. Hosts the solution's Web UI as a Single-page Application (SPA).
Amazon CloudFront	**Supporting**. Delivers the solution's Web UI
Amazon API Gateway	**Supporting**. Creates the solution's REST API to support the user interface.
AWS WAF	**Supporting**. Protects the solution's Web UI.
Amazon Cognito	**Supporting**. Used to authenticate and authorize access to the solution's Web UI.

Plan your deployment

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

Getting started

Time to deploy: Approximately 30 minutes per account.

We recommend adopting the solution in phases rather than enabling automatic remediation everywhere at once. With a phased approach, you build confidence in each remediation against your own environment before the solution acts on findings without human review.

Use a non-production environment first

Before you enable fully automated remediation in production, validate the controls you intend to automate in a non-production account and Region. Reproduce the finding on a disposable resource, run the remediation, and confirm the outcome. This gives you a safe place to experiment with new controls and filters without affecting production workloads.

Walk, then run

1. **Observe.** Deploy the solution and review the findings that arrive in the Web UI. On a fresh deployment, automatic remediation is off by default, so no remediations run automatically at this stage. If instead you are upgrading from an earlier ASR version, any controls you had configured for automatic remediation remain enabled after the upgrade — so remediations may run automatically. In that case, review the **Controls** page (or the Remediation Configuration table) to confirm which controls are enabled before proceeding. Use this phase to confirm that findings from your member accounts and Regions are aggregating to the admin account and appearing in the solution.
2. **Remediate on demand.** Run remediations manually from the Web UI, or from the AWS Security Hub CSPM custom action, on individual findings. Review the remediation result and the affected resource before moving on. Where a remediation supports rollback, you can restore the resource to its pre-remediation state from the Web UI. With on-demand remediation, you validate a control's behavior on real resources without committing to automatic action.
3. **Automate selectively.** When you are confident in a control's behavior, enable fully automated remediation for that control. Enable one control at a time. Use filters to limit the accounts,

organizational units, and resource tags the solution acts on. For the steps, see Manage automated remediation.

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 $14.70 for 300 remediations/month
- Medium deployment (100 accounts, 1 region - US East/N. Virginia): Approximately $106.40 for 3,000 remediations/month
- Large deployment (1,000 accounts, 10 regions): Approximately $7,360.00 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	No 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	No Free Tier	Each `aws:executeScript` action step is charged at $0.00003 for every second.
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 CSPM - Security Checks	No Free Tier	First 100,000 checks/account/Region/month costs $0.0010 per check Next 400,000 checks/account/Region/month costs $0.0008 per check Over 500,000 checks/account/Region/month costs $0.0005 per check

Service	Free Tier	Pricing [USD]
AWS Security Hub CSPM - 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 1 Million API requests (not applicable to GetMetricData, GetInsightRuleReport and GetMetricWidgetImage)	First 10,000 metrics costs $0.30 metric/month Next 240,000 metrics costs $0.10 metric/month Next 750,000 metrics costs $0.05 metric/month Over 1,000,000 metrics costs $0.02 metric/month 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 for each alarm metric High Resolution (10 sec) costs $0.30 per alarm metric Standard Resolution Anomaly Detection costs $0.30 per alarm High Resolution Anomaly Detection costs $0.90 per alarm 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
AWS Lambda - Requests	1M free requests per month	$0.20 per 1M requests
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.

Service	Free Tier	Pricing [USD]
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 Third-party (SaaS) events cost $1.00/million events published 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

Service	Free Tier	Pricing [USD]
Amazon DynamoDB	First 25GB of storage is free	$0.25 per GB per month for storage beyond the free tier. On-demand request pricing is $1.25 per 1 million write request units and $0.25 per 1 million read request units. The solution provisions several tables in the admin account (Findings, Remediation History, Remediation Configuration, Resource Filters, Notification Configuration, Notification Batches, Scheduling, and User Account Mapping). Storage for the Findings and Remediation History tables grows with the number of findings aggregated across your accounts and Regions.

Service	Free Tier	Pricing [USD]
AWS Key Management Service	20,000 requests/month	$1.00 per 1 AWS KMS key. $0.03 per 10,000 API requests. For AWS KMS keys that you rotate automatic ally or on demand, the first and second rotations of the key add $1/month (prorated hourly) in cost. **Note: This solution includes AWS KMS caching optimizat ions (S3 Bucket Keys, 60-minute SQS data key reuse, 5-minute Secrets Manager caching) that reduce AWS KMS API calls by approxima tely 70%.**
Amazon Cognito	In the Essentials tier, the first 10,000 Monthly Active Users are free. Note: This free tier is 50 Monthly Active Users when users authenticate via external IdP (SAML/OIDC).	$0.015 per Monthly Active User greater than 10,000 users.
Amazon CloudFront	Free tier includes 1 TB of data transfer out and 10,000,000 HTTP or HTTPS Requests per month.	(US/Canada/Mexico) First 9TB is $0.085 per month. Next 40TB is $0.080 per month. $0.0075 per HTTP request. $0.0100 per HTTPS request.

Service	Free Tier	Pricing [USD]
Amazon S3	No Free Tier	First 50 TB is $0.023 per GB per month. $0.005 per 1,000 PUT, COPY, POST, LIST requests. $0.0004 per 1,000 GET, SELECT, and all other requests.
Amazon API Gateway	1 Million REST API calls in the first 12 months of usage.	$3.50 per million for the first 333 million API calls.
AWS Secrets Manager	30-day free trial per secret.	$0.40 per secret per month. $0.05 per 10,000 API calls. Used to store credentials for the Slack, JIRA, and ServiceNow notification and ticketing channels; one secret per configured channel.
AWS WAF	No Free Tier	$5.00 per web ACL per month. $1.00 per rule per month. $0.60 per 1 million requests. Deployed to protect the Web UI API when the Web UI is enabled, with 10 rules (7 AWS Managed Rule groups plus 3 rate-based rules for per-user and per-IP rate limiting). The AWS Managed Bot Control rule group adds intelligent threat mitigation pricing of $10.00 per month plus $1.00 per 1 million requests.

Service	Free Tier	Pricing [USD]
AWS CloudTrail	First copy of management events is free.	$2.00 per 100,000 management events. Data events cost $0.10 per 100,000 events. Only incurred when you enable the Action Log with the **EnableCloudTrailForASRActionLog** parameter.

AWS KMS cost optimization

Since **version 3.1.0**, this solution includes AWS KMS caching optimizations that reduce cryptographic operation costs by approximately 70%

- **S3 Bucket Keys**: Reduces AWS KMS GenerateDataKey calls for S3 encryption operations
- **SQS Data Key Reuse**: 60-minute cache period for message encryption
- **Secrets Manager Caching**: 5-minute TTL in Lambda functions

Performance Impact: These optimizations improve latency by 10-15ms for S3 operations and full workflows while reducing costs, with no throughput degradation.

Pricing examples (monthly)

Note

The Amazon DynamoDB figures in the following examples account for the Findings and Remediation History tables, which dominate storage. The solution's other tables — Remediation Configuration, Resource Filters, Notification Configuration, Notification Batches, Scheduling, and User Account Mapping — hold negligible data and are omitted for clarity. The examples also assume you have not configured Slack, JIRA, or ServiceNow channels; each channel you configure adds one AWS Secrets Manager secret (see the preceding component table). The AWS Lambda and Amazon SNS line items include the notification pipeline (the notification-dispatcher and per-channel fanout Lambda functions and the per-configuration SNS topics they publish to); at the request volumes modeled here these add well under $0.01 to Lambda ($0.20 per 1M requests) and SNS ($0.50 per 1M

requests). AWS WAF (and its Bot Control intelligent threat mitigation charge) appears only in the Web UI-enabled examples, since the WAF-protected API Gateway is deployed only when the Web UI is enabled.

Example 1: 300 remediations per month

- 10 accounts, 1 Region
- 30 remediations per account/Region/month
- 500 Security Hub findings processed per account/Region/month
- **Web UI disabled**
- **Action Log disabled**
- Total cost $14.70 per month

Service	Assumptions	Monthly charges [USD]
AWS Systems Manager Automation	Steps: ~4 steps * 300 remediations * $0.002 = $2.40 Duration: 10s * 300 remediations * $0.00003 = $0.09	$2.49
AWS Security Hub	No billable services utilized	$0
Amazon CloudWatch Logs	$0.50 per GB	< $0.01
AWS Lambda - Requests	300 remediations * 7 requests = 2,100 requests 5,000 findings * 1 request = 5,000 requests $0.20 / 1,000,000 requests = $0.0000002 per request	$0.00142
AWS Lambda - Duration	(512MB Memory)	$0.029

Service	Assumptions	Monthly charges [USD]
	4,000ms * 300 remediations * $0.0000000083 = $0.00996 449ms * 5,000 findings * $0.0000000083 = $.0186	
AWS Step Functions	19 state transitions * 300 remediations = 5,700 $0.025 * (5,700/1,000) state transitions = $0.14	$0.14
Amazon EventBridge rules	No charge for rules	$0
AWS Key Management Service	1 key * 10 accounts * 1 Region * $1 = $10 (Encrypt/Decrypt API requests) (300 remediations * 2 requests) + (5,000 findings * 4 requests) = 20,600 requests With AWS KMS caching: 20,600 * 0.30 = 6,180 requests $0.03 per 10,000 requests ⇒ $0.03 * (6,180 / 10,000) = $0.02	$10.02

Service	Assumptions	Monthly charges [USD]
Amazon DynamoDB	$2.00 * 1,000,000 read and writes = $2.00 (Findings Table) 15MB * 10 accounts * 1 region = 150MB (History Table) 10MB * 10 accounts * 1 region = 100MB $0.25 per GB-month * 0.25 GB = $0.0625	$2.0625
Amazon SQS	$0.40 * 1,000,000 requests = $0.40	$0.40
Amazon SNS	$0.50 * (600 / 1,000,000 notifications) = $0.0003	$0.0003
Amazon CloudWatch - Metrics	(Enhanced Metrics Disabled) $0.30 * 7 custom metrics = $2.10 $0.01 * (300 put metrics API calls / 1,000) = $0.003	$2.10
Amazon CloudWatch - Dashboards	$3.00 * 1 dashboard = $3.00	$3.00
Amazon CloudWatch - Alarms	(Enhanced Metrics Disabled) $0.10 * 4 alarms = $0.40	$0.40

Service	Assumptions	Monthly charges [USD]
Amazon CloudWatch - X-Ray Traces	300 remediations * 7 requests = 2,100 Lambda invocations 5,000 findings * 1 request = 5,000 Lambda invocations $0.000005 per trace * 7,100 traces = $0.0355	$0.0355
Total		**$14.70**

Example 2: 300 remediations per month (Web UI Enabled)

- 10 accounts, 1 Region
- 30 remediations per account/Region/month
- 5,000 Security Hub findings processed per account/Region/month
- **Web UI enabled**
- **Action Log disabled**
- Total cost $49.31 per month

Service	Assumptions	Monthly charges [USD]
AWS Systems Manager Automation	Steps: ~4 steps * 300 remediations * $0.002 = $2.40 Duration: 10s * 300 remediations * $0.00003 = $0.09	$2.49
AWS Security Hub	No billable services utilized	$0
Amazon CloudWatch Logs	$0.50 per GB	< $0.01

Service	Assumptions	Monthly charges [USD]
AWS Lambda - Requests	300 remediations * 7 requests = 2,100 requests 5,000 findings * 1 request = 5,000 requests $0.20 / 1,000,000 requests = $0.0000002 per request	$0.00142
AWS Lambda - Duration	(512MB Memory) 4,000ms * 300 remediations * $0.0000000083 = $0.00996 449ms * 5,000 findings * $0.0000000083 = $.0186	$0.029
AWS Step Functions	19 state transitions * 300 remediations = 5,700 $0.025 * (5,700/1,000) state transitions = $0.14	$0.14
Amazon EventBridge rules	No charge for rules	$0

Service	Assumptions	Monthly charges [USD]
AWS Key Management Service	1 key * 10 accounts * 1 Region * $1 = $10 (Encrypt/Decrypt API requests) (300 remediations * 2 requests) + (5,000 findings * 4 requests) = 20,600 requests With AWS KMS caching: 20,600 * 0.30 = 6,180 requests $0.03 per 10,000 requests ⇒ $0.03 * (6,180 / 10,000) = $0.02	$10.02
Amazon DynamoDB	$2.00 * 1,000,000 read and writes = $2.00 (Findings Table) 15MB * 10 accounts * 1 region = 150MB (History Table) 10MB * 10 accounts * 1 region = 100MB $0.25 per GB-month * 0.25 GB = $0.0625	$2.0625
Amazon SQS	$0.40 * 1,000,000 requests = $0.40	$0.40
Amazon SNS	$0.50 * (600 / 1,000,000 notifications) = $0.0003	$0.0003

Service	Assumptions	Monthly charges [USD]
Amazon CloudWatch - Metrics	(Enhanced Metrics Disabled) $0.30 * 7 custom metrics = $2.10 $0.01 * (300 put metrics API calls / 1,000) = $0.003	$2.10
Amazon CloudWatch - Dashboards	$3.00 * 1 dashboard = $3.00	$3.00
Amazon CloudWatch - Alarms	(Enhanced Metrics Disabled) $0.10 * 4 alarms = $0.40	$0.40
Amazon CloudWatch - X-Ray Traces	300 remediations * 7 requests = 2,100 Lambda invocations 5,000 findings * 1 request = 5,000 Lambda invocations $0.000005 per trace * 7,100 traces = $0.0355	$0.0355
Amazon Cognito	(Essentials Tier) 500 Monthly Active Users	$0
Amazon CloudFront	Regional Data Transfer Out to Origin (per GB) = $0.020 Regional Data Transfer Out to Internet (per GB) = $0.085 Request Pricing for All HTTP Methods (per 10,000) = $0.0075	$0.1125

Service	Assumptions	Monthly charges [USD]
Amazon S3	(UI Hosting) $0.023 per GB * 0.002 GB = $0.000046 (History Export) $0.023 per GB * 0.50 GB = $0.0125 $0.0004 per 1,000 GET requests	$0.0125
AWS WAF	1 Web ACL = $5.00 per month 10 rules * $1.00 per rule = $10.00 (7 AWS Managed Rule groups + 3 rate-based rules) Bot Control intelligent threat mitigation = $10.00 per month (plus $1.00 per 1M requests and $0.60 per 1M WAF request charges, both negligible at Web UI request volumes)	$25.00
Amazon API Gateway	$3.50 per million REST API calls	$3.50
Total		**$49.31**

Example 3: 3,000 remediations per month

- 100 accounts, 1 Region
- 30 remediations per account/Region/month
- 500 Security Hub findings processed per account/Region/month
- **Web UI disabled**

- **Action Log disabled**
- Total cost $106.49 per month

Service	Assumptions	Monthly charges [USD]
AWS Systems Manager Automation	Steps: ~4 steps * 3,000 remediations * $0.002 = $24.00 Duration: 10s * 3,000 remediations * $0.00003 = $0.90	$24.90
AWS Security Hub	No billable services utilized	$0
Amazon CloudWatch Logs	$0.50 per GB	< $0.01
AWS Lambda - Requests	3,000 remediations * 7 requests = 21,000 requests 50,000 findings * 1 request = 50,000 requests $0.20 / 1,000,000 requests = $0.0000002 per request	$0.01
AWS Lambda - Duration	(512MB Memory) 4,000ms * 3,000 remediations * $0.0000000083 = $0.0996 449ms * 50,000 findings * $0.0000000083 = $0.186	$0.29
AWS Step Functions	19 state transitions * 3,000 remediations = 57,000 $0.025 * (57,000/1,000) state transitions = $1.425	$1.425

Service	Assumptions	Monthly charges [USD]
Amazon EventBridge rules	No charge for rules	$0
AWS Key Management Service	1 key * 100 accounts * 1 Region * $1 = $100 (Encrypt/Decrypt API requests) (3,000 remediations * 2 requests) + (50,000 findings * 4 requests) = 206,000 requests With AWS KMS caching: 206,000 * 0.30 = 61,800 requests $0.03 per 10,000 requests ⇒ $0.03 * (61,800 / 10,000) = $0.185	$100.185
Amazon DynamoDB	$2.00 * 1,000,000 read and writes = $2.00 (Findings Table) 15MB * 100 accounts * 1 region = 1,500MB (History Table) 10MB * 100 accounts * 1 region = 1,000MB $0.25 per GB-month * 2.5 GB = $0.625	$2.625
Amazon SQS	$0.40 * 1,000,000 requests = $0.40	$0.40

Service	Assumptions	Monthly charges [USD]
Amazon SNS	$0.50 * 1,000,000 notifications = $0.50	$0.50
Amazon CloudWatch - Metrics	(Enhanced Metrics Disabled) $0.30 * 7 custom metrics = $2.10 $0.01 * (3000 / 1,000) put metrics API calls = $0.03	$2.13
Amazon CloudWatch - Dashboards	$3.00 * 1 dashboard = $3.00	$3.00
Amazon CloudWatch - Alarms	$0.10 * 4 alarms = $0.40	$0.40
Amazon CloudWatch - X-Ray Traces	3,000 remediations * 7 requests = 21,000 Lambda invocations 50,000 findings * 1 request = 50,000 Lambda invocations $0.000005 per trace * 71,000 traces = $0.355	$0.355
Total		**$106.49**

Example 4: 30,000 remediations per month

- 1,000 accounts, 10 Regions
- 30 remediations per account/Region/month
- 500 Security Hub findings processed per account/Region/month
- **Web UI disabled**
- **Action Log disabled**
- Total cost $7,360.00 per month

Service	Assumptions	Monthly charges [USD]
AWS Systems Manager Automation	Steps: ~4 steps * 30,000 remediations * $0.002 = $240.00 Duration: 10s * 30,000 remediations * $0.00003 = $9.00	$249.00
AWS Security Hub	No billable services utilized	$0
Amazon CloudWatch Logs	$0.50 per GB	< $0.01
AWS Lambda - Requests	30,000 remediations * 7 requests = 210,000 requests 5,000,000 findings * 1 request = 5,000,000 requests $0.20 / 1,000,000 requests = $0.0000002 per request	$1.042
AWS Lambda - Duration	(512MB Memory) 4,000ms * 30,000 remediations * $0.0000000083 = $0.996 449ms * 5,000,000 findings * $0.0000000083 = $18.63	$19.63
AWS Step Functions	19 state transitions * 30,000 remediations = 570,000 $0.025 * (570,000/1,000) state transitions = $14.25	$14.25
Amazon EventBridge rules	No charge for rules	$0

Service	Assumptions	Monthly charges [USD]
AWS Key Management Service	(1 key) $1 * 1,000 accounts * 10 Region = $10,000 (Encrypt/Decrypt API requests) (30,000 remediations * 2 requests) + (5,000,000 findings * 4 requests) = 20,060,000 requests With AWS KMS caching: 20,060,000 * 0.30 = 6,018,000 requests $0.03 per 10,000 requests ⇒ $0.03 * (6,018,000 / 10,000) = $18.05	$10,018.05
Amazon DynamoDB	$2.00 * (10,000,000 read and writes / 1,000,000) = $20.00 (Findings Table) 15MB * 1000 accounts * 10 region = 150GB (History Table) 10MB * 1000 accounts * 10 region = 100GB $0.25 per GB-month * 250 GB = $62.50	$82.50
Amazon SQS	$0.40 * (5,060,000 requests / 1,000,000) = $2.024	$2.024
Amazon SNS	$0.000005 * 1,000,000 notifications = $0.50	$0.50

Service	Assumptions	Monthly charges [USD]
Amazon CloudWatch - Metrics	(Enhanced Metrics Disabled) $0.30 * 7 custom metrics = $2.10 $0.01 * (30,000 / 1,000) put metrics API calls = $0.30	$2.40
Amazon CloudWatch - Dashboards	$3.00 * 1 dashboard = $3.00	$3.00
Amazon CloudWatch - Alarms	(Enhanced Metrics Disabled) $0.10 * 4 alarms = $0.40	$0.40
Amazon CloudWatch - X-Ray Traces	30,000 remediations * 7 requests = 210,000 Lambda invocations 5,000,000 findings * 1 request = 5,000,000 Lambda invocations $0.000005 per trace * 5,210,000 traces = $26.05	$26.05
Total		**$7,360.00**

Example 5: 30,000 remediations per month (Web UI Enabled)

- 1,000 accounts, 10 Regions
- 30 remediations per account/Region/month
- 500 Security Hub findings processed per account/Region/month
- **Web UI enabled**
- **Action Log disabled**
- Total cost $7,393.10 per month

Service	Assumptions	Monthly charges [USD]
AWS Systems Manager Automation	Steps: ~4 steps * 30,000 remediations * $0.002 = $240.00 Duration: 10s * 30,000 remediations * $0.00003 = $9.00	$249.00
AWS Security Hub	No billable services utilized	$0
Amazon CloudWatch Logs	$0.50 per GB	< $0.01
AWS Lambda - Requests	30,000 remediations * 7 requests = 210,000 requests 5,000,000 findings * 1 request = 5,000,000 requests $0.20 / 1,000,000 requests = $0.0000002 per request	$1.042
AWS Lambda - Duration	(512MB Memory) 4,000ms * 30,000 remediations * $0.0000000083 = $0.996 449ms * 5,000,000 findings * $0.0000000083 = $18.63	$19.63
AWS Step Functions	19 state transitions * 30,000 remediations = 570,000 $0.025 * (570,000/1,000) state transitions = $14.25	$14.25
Amazon EventBridge rules	No charge for rules	$0

Service	Assumptions	Monthly charges [USD]
AWS Key Management Service	(1 key) $1 * 1,000 accounts * 10 Region = $10,000 (Encrypt/Decrypt API requests) (30,000 remediations * 2 requests) + (5,000,000 findings * 4 requests) = 20,060,000 requests With AWS KMS caching: 20,060,000 * 0.30 = 6,018,000 requests $0.03 per 10,000 requests ⇒ $0.03 * (6,018,000 / 10,000) = $18.05	$10,018.05
Amazon DynamoDB	$2.00 * (10,000,000 read and writes / 1,000,000) = $20.00 (Findings Table) 15MB * 1000 accounts * 10 region = 150GB (History Table) 10MB * 1000 accounts * 10 region = 100GB $0.25 per GB-month * 250 GB = $62.50	$82.50
Amazon SQS	$0.40 * (5,060,000 requests / 1,000,000) = $2.024	$2.024
Amazon SNS	$0.000005 * 1,000,000 notifications = $0.50	$0.50

Service	Assumptions	Monthly charges [USD]
Amazon CloudWatch - Metrics	(Enhanced Metrics Disabled) $0.30 * 7 custom metrics = $2.10 $0.01 * (30,000 / 1,000) put metrics API calls = $0.30	$2.40
Amazon CloudWatch - Dashboards	$3.00 * 1 dashboard = $3.00	$3.00
Amazon CloudWatch - Alarms	(Enhanced Metrics Disabled) $0.10 * 4 alarms = $0.40	$0.40
Amazon CloudWatch - X-Ray Traces	30,000 remediations * 7 requests = 210,000 Lambda invocations 5,000,000 findings * 1 request = 5,000,000 Lambda invocations $0.000005 per trace * 5,210,000 traces = $26.05	$26.05
Amazon Cognito	(Essentials Tier) 5,000 Monthly Active Users	$0

Service	Assumptions	Monthly charges [USD]
Amazon CloudFront	Regional Data Transfer Out to Origin (per GB) = $0.020 Regional Data Transfer Out to Internet (per GB) = $0.085 Request Pricing for All HTTP Methods (per 10,000) = $0.0075	$0.1125
Amazon S3	(UI Hosting) $0.023 per GB * 0.002 GB = $0.000046 (History Export) $0.023 per GB * 100 GB = $2.30 $0.0004 per 1,000 GET requests * 5,000 requests = $2.00	$4.30
AWS WAF	1 Web ACL = $5.00 per month 10 rules * $1.00 per rule = $10.00 (7 AWS Managed Rule groups + 3 rate-based rules) Bot Control intelligent threat mitigation = $10.00 per month (plus $1.00 per 1M requests and $0.60 per 1M WAF request charges, both negligible at Web UI request volumes)	$25.00

Service	Assumptions	Monthly charges [USD]
Amazon API Gateway	$3.50 per million REST API calls	$3.50
Total		**$7,393.10**

Important

AWS KMS Key Rotation Costs AWS Key Management Service (AWS 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 3,750 * $0.0000166667 = $0.0625 compute time cost 0.15 * $0.20 = $0.03 request cost $0.0625 + $0.03 = $0.0925 total Lambda cost	$0.0925
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.

API Gateway Security Policy

If you choose to enable the solution's Web User Interface, an API Gateway REST API is deployed alongside the Admin CloudFormation stack which serves as the backend for all operations in the Web UI. The REST API deployed by the solution uses the default TLS security policy for API Gateway, which is `TLS-1-2` for regional APIs.

However, after deploying the Admin CloudFormation stack you may choose to customize the solution's REST API by adding a more restrictive TLS security policy. For example, you can choose the `TLS_1_2 security policy` to restrict for traffic using TLSv1.2 or TLSv1.3. You can find the solution's REST API in the API Gateway console under the name **AutomatedSecurityResponseApi**.

In order to choose a security policy for the solution's REST API, you must first configure a custom domain name. For more information, see Custom domain name for public REST APIs in API Gateway.

For more information on adding a security policy to your REST API, see Choose a security policy for your REST API custom domain in API Gateway in the API Gateway guide.

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

Important

Enabling optional features in the solution may reduce the list of regions supported for deployment. In other words, the list below only applies to the core components of the solution. For example, if you choose to enable the Web UI, you will not be able to deploy the solution in GovCloud regions since CloudFront is not supported in GovCloud (US).

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

Region name	Region code
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
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

Region name	Region code
Asia Pacific (Thailand)	ap-southeast-7
Asia Pacific (Malaysia)	ap-southeast-5
Asia Pacific (Taipei)	ap-east-2
Asia Pacific (New Zealand)	ap-southeast-6

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*.

Amazon CloudWatch quotas

Your AWS account has Amazon CloudWatch quotas tied to CloudWatch Resource Policies, which allow only 10 resource policies per Region per account and cannot be increased. For more information, see Amazon CloudWatch Logs Quotas in the *Amazon CloudWatch User Guide*. Before your deployment, check your current usage to ensure you won't cross this threshold when deploying the solution.

AWS Organizations

The solution's Lambda functions make calls to the AWS Organizations API in order to fetch the alias of the current account to include in messages published to the solution's SNS topic. This enables human-readable account names to be visible in the solution's notifications for debugging and tracking purposes.

AWS Organizations imposes limits on how often customers can invoke their API endpoints. If you find that the solution is exceeding the limits set for your account, you can disable the feature that fetches and displays the account alias.

To do this, **navigate to the Lambda function** named `SO0111-ASR-sendNotifications` located in the region and account where you deployed the Admin stack. Then, **locate the environment variable** named `DISABLE_ACCOUNT_ALIAS_LOOKUP` and change the value from "False" to **"True"**. The account alias field in the solution's notifications will now be *"Unknown"* however this will not impact the functionality of the solution.

Solution limits and defaults

In addition to the AWS service quotas described above, the solution applies its own default limits that are set in the AWS CDK configuration. These are configurable and sit well below the underlying AWS service quotas. They exist as safety floors to contain unexpected load or runaway loops, not as tight per-user quotas. The tables below list the values that differ from the corresponding AWS service default.

Limit	Solution default	AWS service default
Amazon API Gateway stage throttle (account-wide)	500 requests/second steady, 1,000 burst	10,000 requests/second, 5,000 burst

Limit	Solution default	AWS service default
AWS WAF per-user rate limit (60-second window)	1,000 requests	No default rule
AWS WAF per-IP rate limit (60-second window)	2,000 requests	No default rule
AWS WAF sensitive-write rate limit (60-second window)	300 requests	No default rule

Note

The per-user rate limits aggregate on the caller's access token; requests without one are covered by the per-IP rule. Requests that exceed an AWS WAF rate-based rule are blocked and receive an HTTP 429 (Too Many Requests) response with a JSON body.

The solution also configures the following CloudWatch alarm thresholds for observability. These alarms notify operators; they do not block requests.

Alarm	Threshold
Control state changes	5 per minute
Sensitive writes	20 per minute

The following compute and processing defaults apply:

Limit	Solution default	AWS service default
Orchestrator (AWS Step Functions) execution timeout	23 hours	1 year (maximum)
Notification dispatcher Lambda function memory	256 MB	128 MB

Limit	Solution default	AWS service default
Notification dispatcher Lambda function reserved concurrency	10	None (unreserved)
Notification dispatcher Lambda function timeout	30 seconds (60 seconds for webhook delivery)	3 seconds
Amazon SQS message retention	14 days	4 days
AWS KMS data key reuse period	60 minutes	5 minutes

The solution sets the following data retention (Amazon DynamoDB TTL and Amazon S3) and export limits:

Item	Solution default
Findings retention	8 days
Remediation History retention	365 days
Export file retention	30 days
Pre-signed URL validity	1 day
Maximum export time per request	26,000 milliseconds
Maximum records per export	50,000

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 CSPM, refer to Setting up AWS Security Hub CSPM in the *AWS Security Hub User Guide.* This solution also supports AWS Security Hub (non-CSPM version). For more information about setting up AWS Security Hub, refer to Enabling Security Hub.

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 AWS Security Hub, only enable the Security Control (SC) 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. Consolidated control findings is enabled by default if you enable Security Hub CSPM on or after February 23, 2023.

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.

Important

If you are using AWS Security Hub (non-CSPM) then you are responsible for ensuring your member accounts onboarded with AWS Security Hub CSPM are also onboarded with AWS Security Hub (non-CSPM). Regions aggregated in AWS Security Hub CSPM should also match regions aggregated in AWS Security Hub (non-CSPM).

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 for each account. It does not matter in which Region you deploy the member roles stack, so for simplicity, deploy the member roles stack to the same Region in which the Admin stack is deployed.

Using the previous example, deploy 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, used 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. Deploy 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**
>
> When this feature is enabled, you must use solution version 2.0.0 or later and enable the "SC" (Security Control) playbook in both the Admin and Member stacks. These stacks deploy the automation documents needed to work with consolidated control IDs. You

do not need to deploy stacks for individual standards (such as AWS FSBP) when using consolidated control findings.

China deployment

The solution does support deployment in China regions, however **you must use the following Launch buttons for one-click deployment in China regions, rather than the Launch buttons provided in other sections of this guide.** Using the "Launch Solution" buttons provided in upcoming sections in this guide will not work if you are deploying in China regions. You can still download the templates from any S3 bucket link and deploy the stacks by uploading the template file.

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

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

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

GovCloud (US) deployment

The solution does support deployment in GovCloud (US) regions, however **you must use the following Launch buttons for one-click deployment in GovCloud (US) regions, rather than the Launch buttons provided in other sections of this guide.** Using the "Launch Solution" buttons provided in upcoming sections in this guide will not work if you are deploying in GovCloud (US) regions. You can still download the templates from any S3 bucket link and deploy the stacks by uploading the template file.

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

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

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

AWS CloudFormation templates

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.

automated-security-response-webui-nested-stack.template - Creates the resources to support the solution's Web UI.

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

View template

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 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.

This solution works with both AWS Security Hub and AWS Security Hub CSPM.

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 because they use nested stacks, 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. Choose `Yes` if `SO0111-ASR-Orchestrator` already exists from a previous installation.
- Choose whether to enable the solution's Web UI. If you choose to enable this feature, you must also enter an email address to be assigned an administrator role.
- Choose your preferences for collecting CloudWatch metrics related to the solution's operational health.

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.
- Enter a value for the `namespace` which will be used to prevent resource name conflicts with a previous or concurrent deployment in the same account. Enter a string of up to 9 lowercase alphanumeric characters.

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 log group (used by some remediations).
- Enter the account ID of your AWS Security Hub admin account.
- Enter a value for the `namespace` which will be used to prevent resource name conflicts with a previous or concurrent deployment in the same account. Enter a string of up to 9 lowercase alphanumeric characters. This should match the `namespace` value you selected for the Member Roles stack, additionally, the namespace value does not need to be unique per member 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.

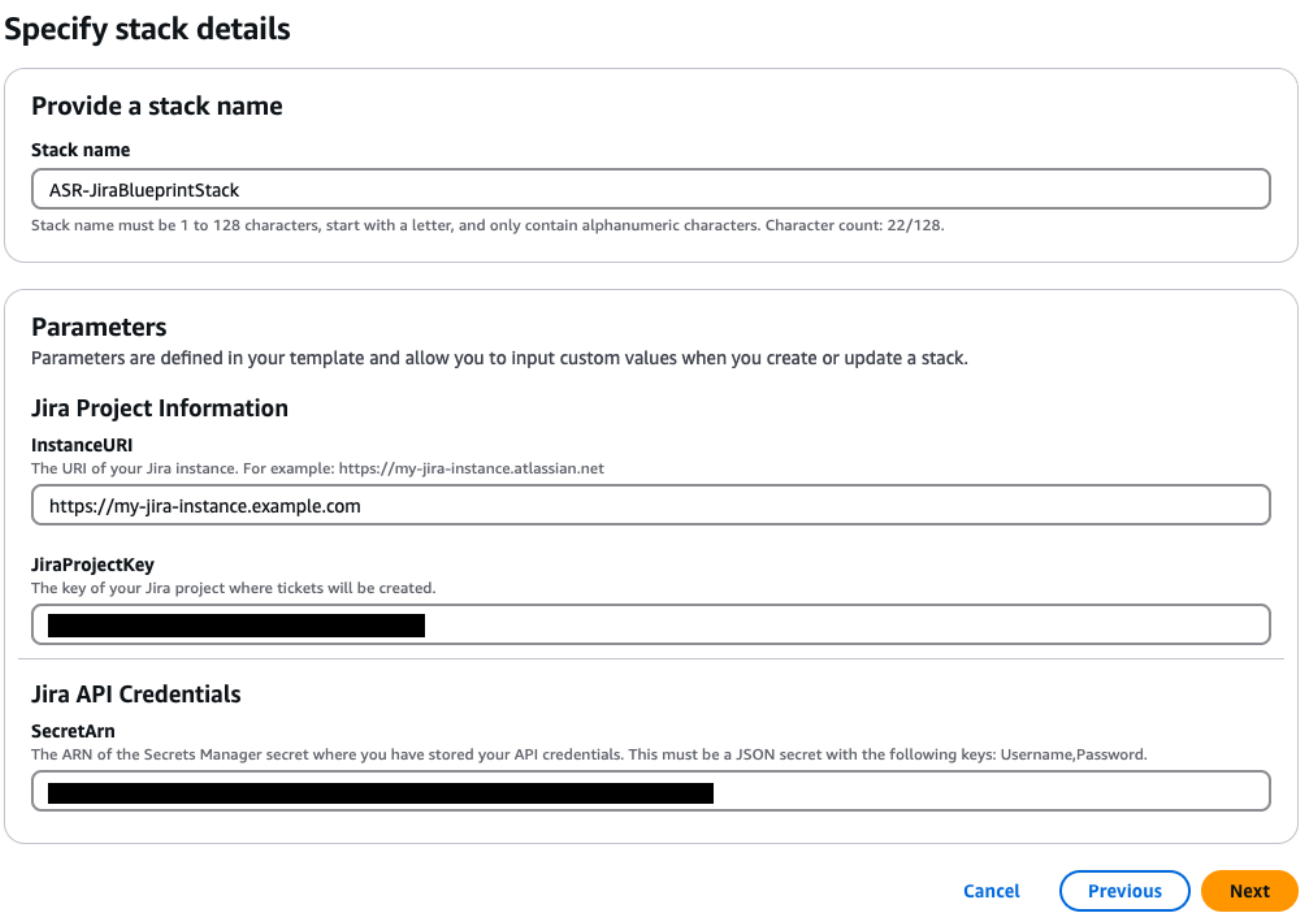
   c. Provide the project key for the Jira project that you want to send tickets to.

   d. Create a new key-value secret in AWS 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.



Jira Field Configuration:

After deploying the Jira stack, you can customize Jira ticket fields by setting the `JIRA_FIELDS_MAPPING` environment variable on the Lambda function. This JSON string overrides default Jira ticket fields and must follow the Jira API fields structure.

Default values when `JIRA_FIELDS_MAPPING` is empty or fields are not specified:

- **priority**: `{"id": "3"}` (Medium priority)
- **issuetype**: `{"id": "10006"}` (Task)
- **accountId**: Automatically retrieved using the `GET /rest/api/2/myself` API endpoint

  Example configuration with custom fields:

```
{
  "reporter": {"accountId": "123456:494dcbff-1b80-482c-a89d-56ae81c145a4"},
  "priority": {"id": "1"},
  "issuetype": {"id": "10006"},
```

```
  "assignee": {"accountId": "123456:another-user-id"},
  "customfield_10001": "custom value"
}
```

Common Jira field IDs:

- **Priority IDs**: 1 (Highest), 2 (High), 3 (Medium), 4 (Low), 5 (Lowest)
- **Issue Type ID**: Varies by Jira project (for example, 10006 for Task)
- **Account ID**: Format `123456:494dcbff-1b80-482c-a89d-56ae81c145a4`

You can find your Jira field IDs and account IDs using the Jira REST API:

- `GET /rest/api/2/myself` for account ID
- `GET /rest/api/2/priority` for priority IDs
- `GET /rest/api/2/project/{projectKey}` for issue type IDs

For more information, refer to the Jira REST API v2 Issue POST format.

To deploy the ServiceNow stack:

f. Enter a name for your stack.

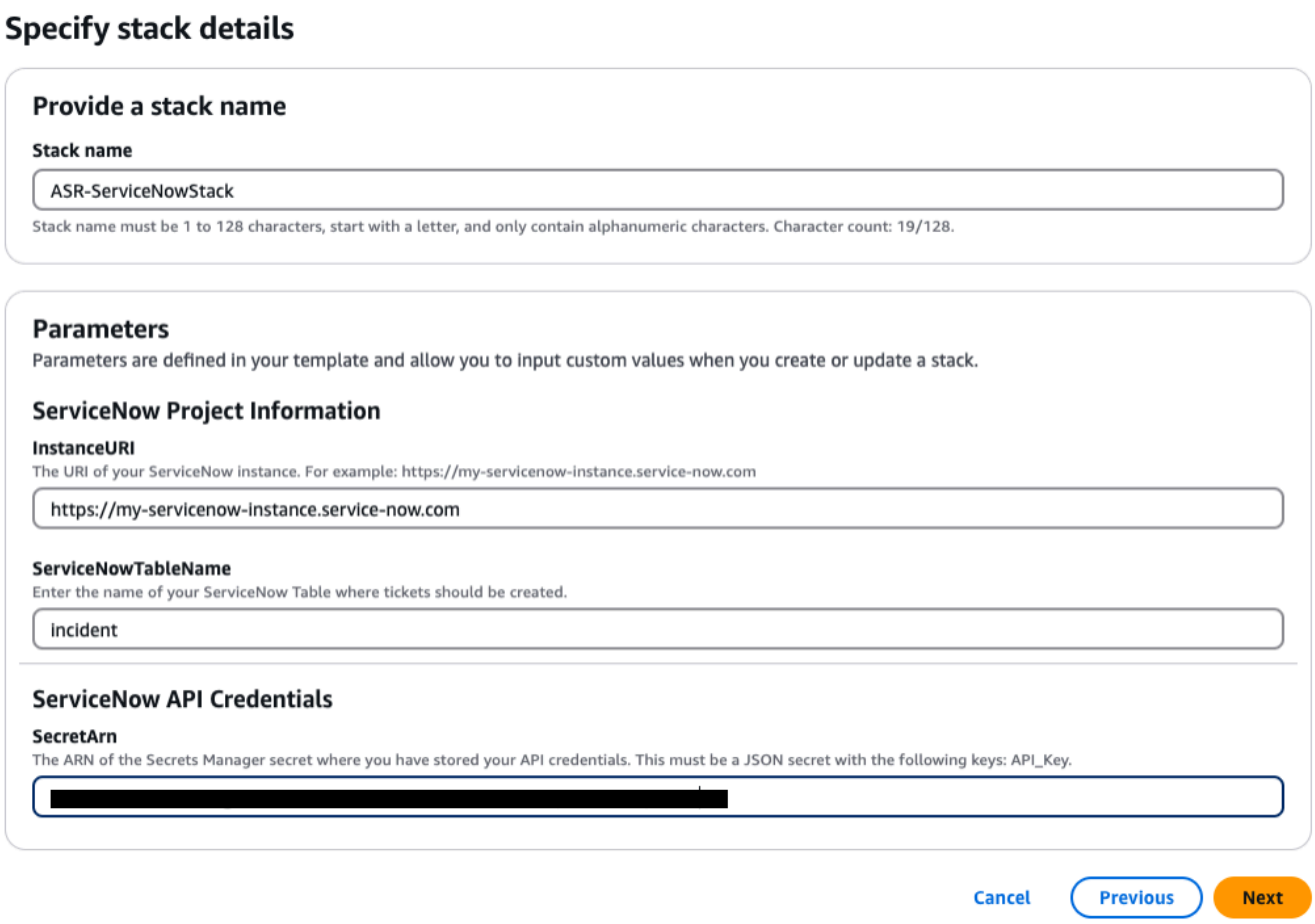
g. Provide the URI of your ServiceNow instance.

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.

Parameters

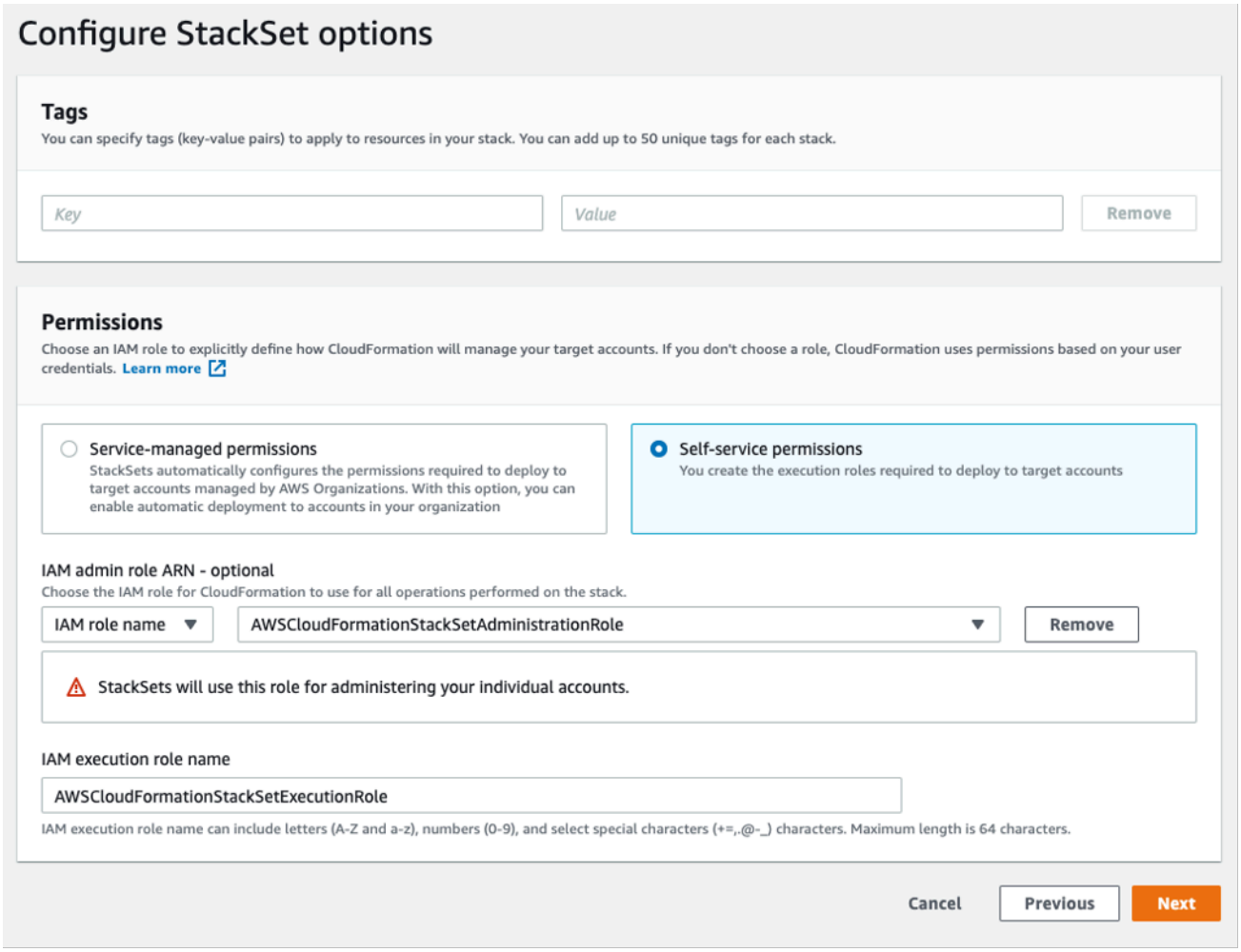
Parameter	Default	Description
Load SC Admin Stack	yes	Specify whether to install the admin components for automated remediation of SC controls.
Load AFSBP Admin Stack	no	Specify whether to install the admin components for automated remediation of FSBP controls.
Load CIS120 Admin Stack	no	Specify whether 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 PCI321 Admin Stack	no	Specify whether to install the admin components for automated remediation of PCI321 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
ShouldDeployWebUI	`yes`	Deploy the Web UI components including API Gateway, Lambda functions , and CloudFront distribut ion. Choose "yes" to enable the web-based user interface for viewing findings and remediation status. If you choose to disable this feature, you can still configure automated remediations and run remediations on-demand using the Security Hub CSPM custom action.

Parameter	Default	Description
AdminUserEmail	*(Optional input)*	Email address for the initial admin user. This user will have full administrative access to the ASR Web UI. Required **only** when Web UI is enabled.
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.
RemediationFailure AlarmThreshold	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. This parameter functions only if alarms are created (see the **Use CloudWatch Metrics Alarms** parameter).

Parameter	Default	Description
EnableEnhancedCloudWatchMetrics	no	If `yes`, creates additional CloudWatch metrics to track all control IDs individually on the CloudWatch dashboard and as CloudWatch alarms. 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` .

Configure StackSet options



1. For the **Account numbers** parameter, enter the account ID of the AWS Security Hub admin account.
2. 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.

   You can view the status of the StackSet operation in the AWS CloudFormation console on the StackSet details page. You should receive a **SUCCEEDED** operation status in approximately 15 minutes.

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.

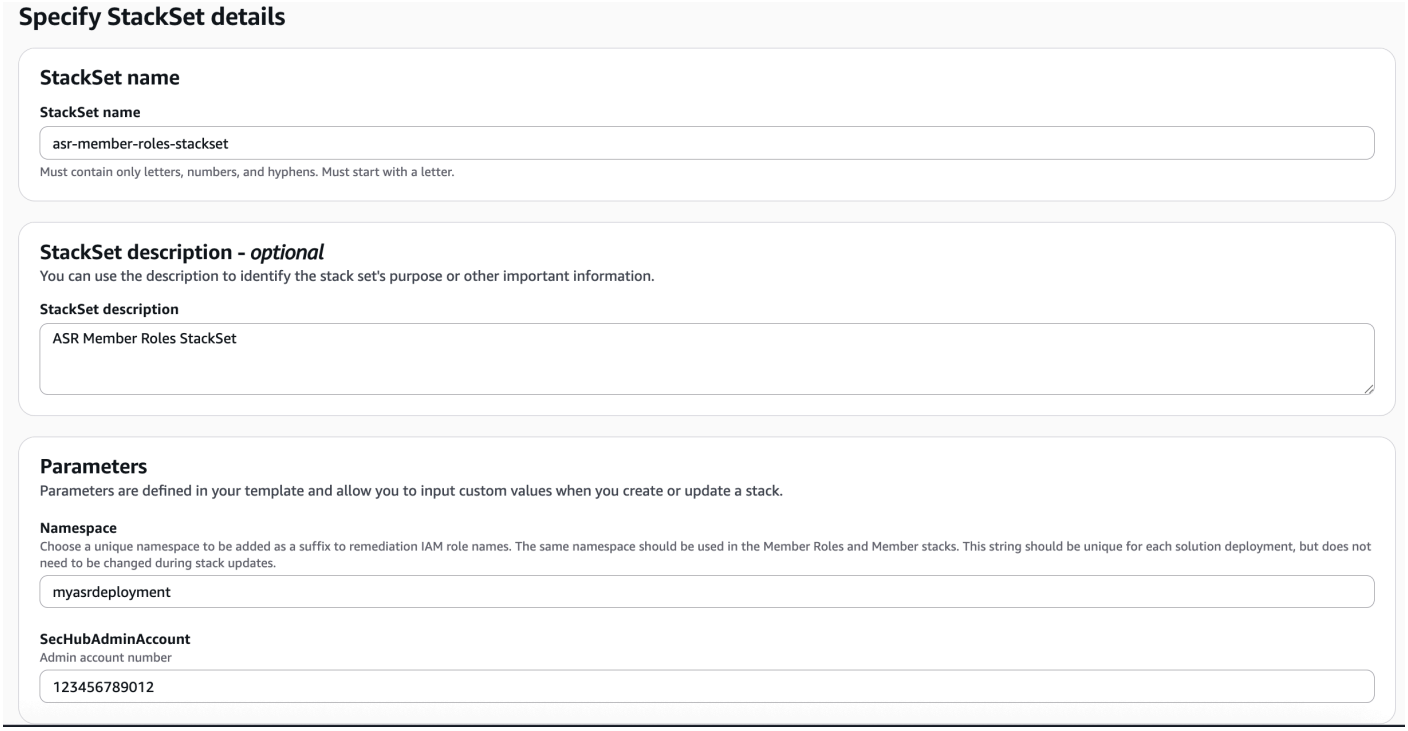
Parameters

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 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 Admin Account	*<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.

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.

You can view the status of the StackSet operation in the AWS CloudFormation console on the StackSet details page. You should receive a **SUCCEEDED** operation status in approximately 5 minutes.

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

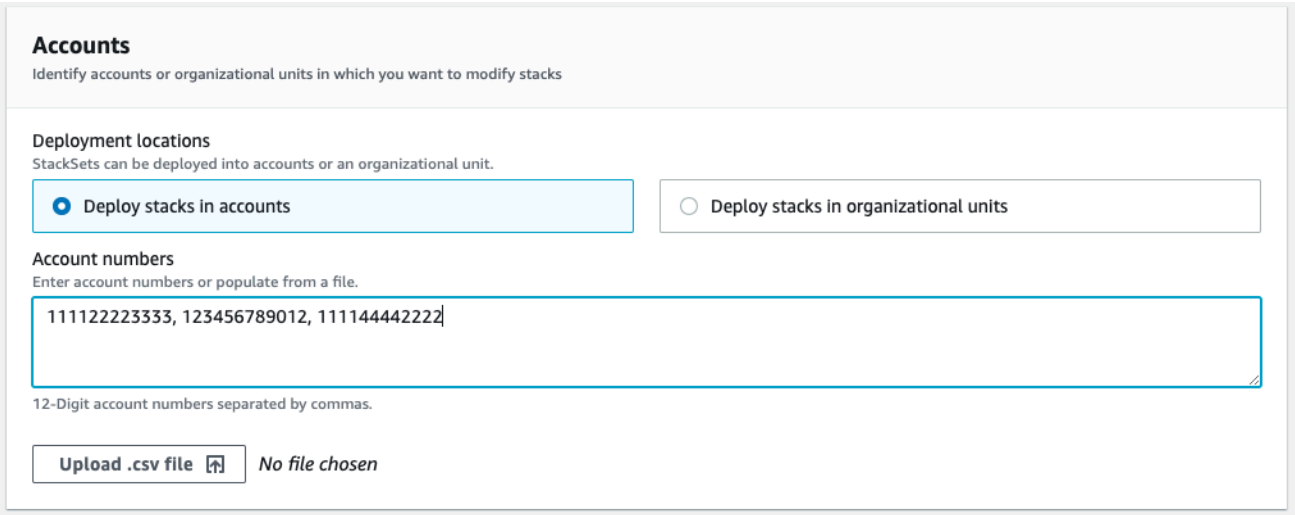
Parameter	Default	Description
Provide the name of the LogGroup to be used to create Metric Filters and Alarms	*<Requires input>*	Specify the name of a CloudWatch Logs group where CloudTrail logs API

Parameter	Default	Description
		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 PCI321 Member Stack	no	Specify whether to install the member components for automated remediation of PCI321 controls.
Load NIST Member Stack	no	Specify whether to install the member components for automated remediation of NIST controls.

Parameter	Default	Description
Create S3 Bucket For Redshift Audit Logging	no	Choose yes to create the S3 bucket 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. **Additionally, you can only enable this feature in a single region within the same account.** See the Cost section to understand the additional cost that this incurs.

Accounts



Deployment locations: You can 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.

You can view the status of the StackSet operation in the AWS CloudFormation console on the StackSet details page. You should receive a **SUCCEEDED** operation status for each account and Region combination. Deployment time varies based on the number of accounts and Regions selected.

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 (choose `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 ID 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 for each account.
- Choose 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.

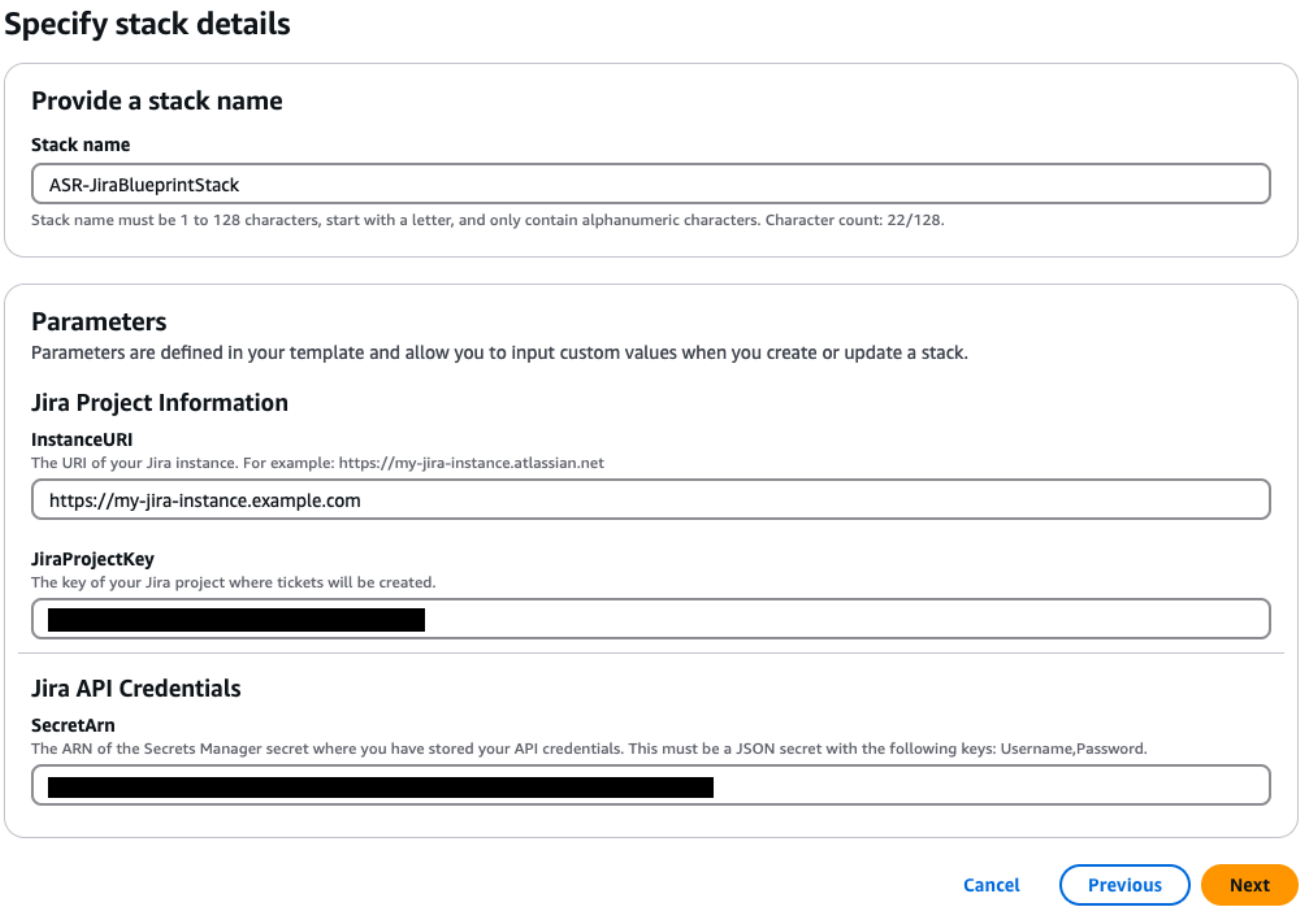
c. Provide the project key for the Jira project that you want to send tickets to.

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.



Jira Field Configuration:

For information on customizing Jira ticket fields, refer to the Jira Field Configuration section in Step 0 of the StackSet deployment.

To deploy the ServiceNow stack:

f. Enter a name for your stack.

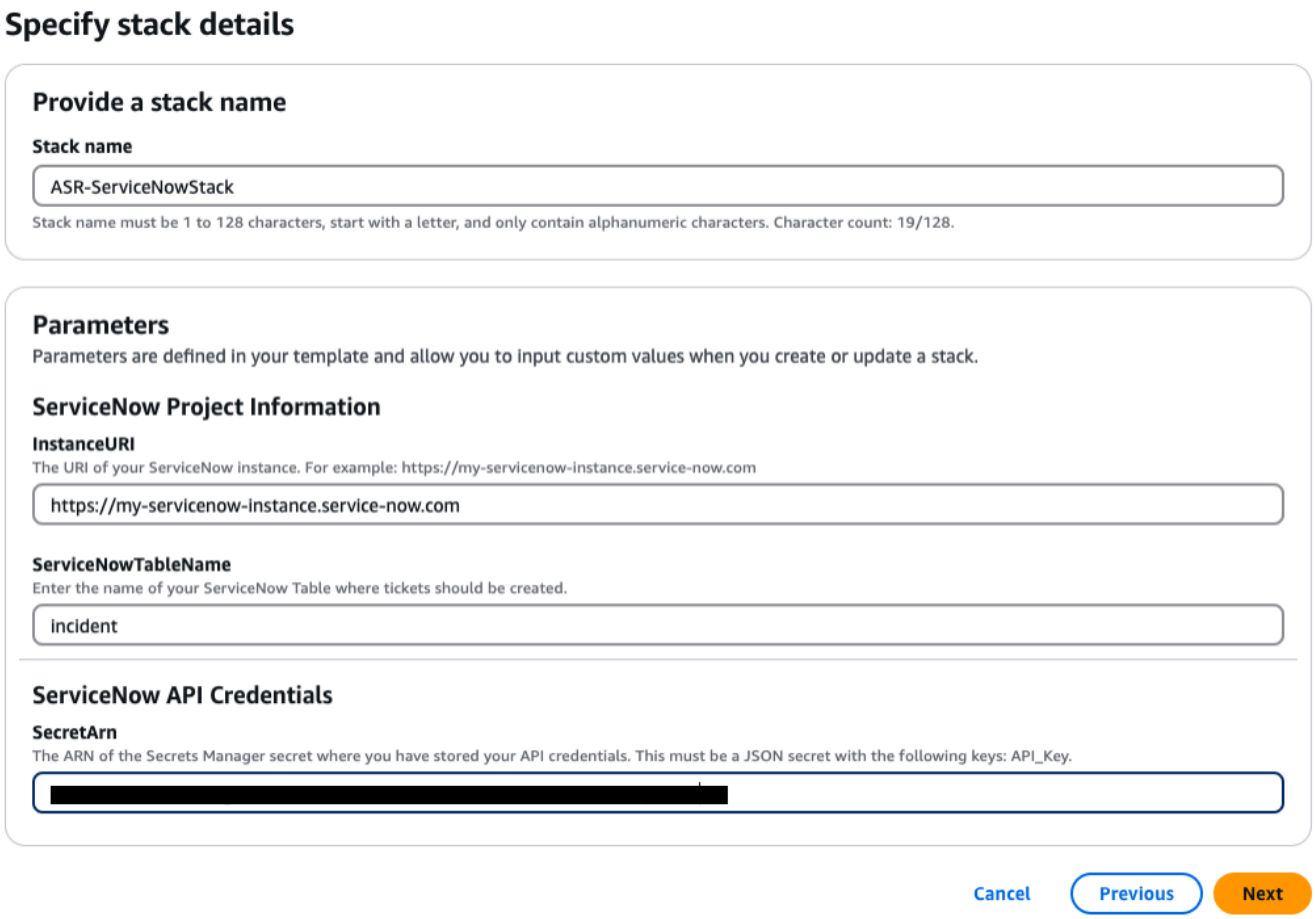
g. Provide the URI of your ServiceNow instance.

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 data collection. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered through this survey. Data collection is subject to the AWS Privacy Notice.

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, see 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 choose the button below to launch the `automated-security-response-admin.template` AWS CloudFormation template.

   Launch solution

   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 PCI321 Admin Stack	no	Specify whether to install the admin components for

Parameter	Default	Description
		automated remediation of PCI321 controls.
Load NIST Admin Stack	no	Specify whether to install the admin components for automated remediation of NIST controls.
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
ShouldDeployWebUI	`yes`	Deploy the Web UI components including API Gateway, Lambda functions , and CloudFront distribut ion. Choose "yes" to enable the web-based user interface for viewing findings and remediation status.

Parameter	Default	Description
AdminUserEmail	*(Optional input)*	Email address for the initial admin user. This user will have full administrative access to the ASR Web UI. Required **only** when Web UI is enabled.
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.
RemediationFailure AlarmThreshold	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. This parameter functions only if alarms are created (see the **Use CloudWatch Metrics Alarms** parameter).

Parameter	Default	Description
EnableEnhancedCloudWatchMetrics	no	If `yes`, creates additional CloudWatch metrics to track all control IDs individually on the CloudWatch dashboard and as CloudWatch alarms. 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`.

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 for each 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). Choose 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 namespace should be used in the Member Roles and Member stacks. This string should be unique for each solution deployment, but

Parameter	Default	Description
		does not need to be changed during stack updates. The namespace value does **not** need to be unique per member account.
Sec Hub Admin Account	*<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 data collection. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered through this survey. Data collection is subject to the AWS Privacy Notice.

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). Choose 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

Parameter	Default	Description
		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 PCI321 Member Stack	no	Specify whether to install the member components for automated remediation of PCI321 controls.
Load NIST Member Stack	no	Specify whether to install the member components for automated remediation of NIST controls.

Parameter	Default	Description
Create S3 Bucket For Redshift Audit Logging	no	Choose `yes` to create the S3 bucket 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. **Additionally, you can only enable this feature in a single region within the same account.** 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.

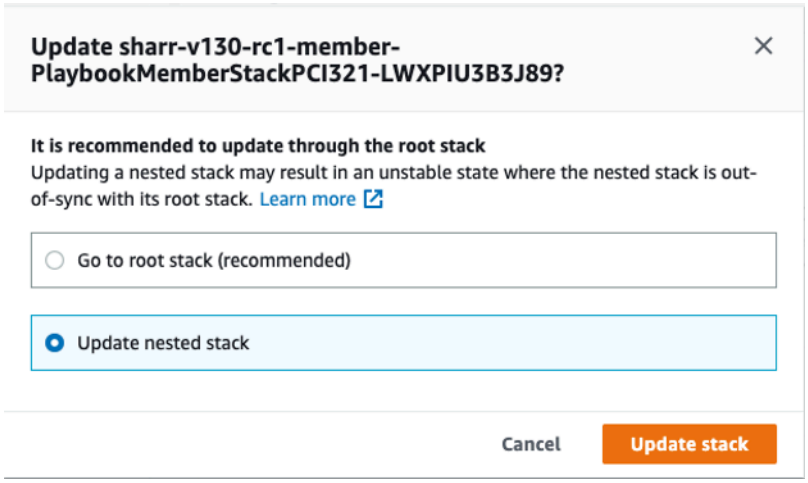
You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a CREATE_COMPLETE status in approximately 15 minutes.

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 solution's 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.

AWS 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, for example, 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, for example, 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 must be encrypted and must 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::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 (for example, asr-staging-us-east-1, asr-staging-eu-west-1, and so on), allowing deployments across multiple Regions without needing to create the buckets in the Management account.

4. Build Preparation

- Prerequisites:
  - AWS CLI v2
  - Python 3.11+ with pip
  - AWS CDK 2.171.1+
  - Node.js 22+ with npm
  - Poetry v2 with plugin to export
- Git clone [automated-security-response-on-aws GitHub repository]

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 (for example, mybucket) and the version you are building (for example, v1.0.0). Use a semver version based on the version downloaded from GitHub (for example, 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/
$SOLUTION_VERSION/ --recursive
aws s3 cp regional-s3-assets/  s3://$ASSET_BUCKET_NAME/$SOLUTION_NAME/
$SOLUTION_VERSION/ --recursive
```

Step 2: Stacks deployment to AWS Control Tower

1. Build manifest for ASR components

After deploying ASR artifacts to the S3 buckets, update the AWS Control Tower pipeline manifest to reference the new version, and then trigger the pipeline run. For more information, see AWS Control Tower deployment.

> **Important**
>
> For 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: ShouldDeployWebUI
        parameter_value: "yes"
      - parameter_key: AdminUserEmail
        parameter_value: "<YOUR EMAIL ADDRESS>"
      - parameter_key: LoadSCAdminStack
        parameter_value: "yes"
```

```
      - parameter_key: LoadCIS120AdminStack
        parameter_value: "no"
      - 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: 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

You can verify the deployment by checking the AWS Control Tower pipeline status in the AWS CodePipeline console.

Using the ASR API programmatically

If you want to integrate ASR into your own systems for example, a security orchestration platform, a ticketing or ChatOps workflow, or a custom dashboard — and drive remediations from there instead of the Web UI, you can call the Automated Security Response on AWS (ASR) API directly. With the ASR API, you can automate and monitor remediations from tooling you already operate.

To do this, use a machine-to-machine (M2M) client with the OAuth 2.0 `client_credentials` grant. When the Web UI is enabled, the solution provisions the Amazon Cognito OAuth scope `asr-api/full-access`. This scope grants the client access to the ASR API. You create the M2M app client yourself after deployment. The solution does not create one, so no machine credential exists until you opt in.

A token with the `asr-api/full-access` scope has **Full Access** — the same capability as the solution administrator role, including the ability to manage other users. Treat the credentials with the same care as administrator sign-in credentials.

Prerequisites

Deploy the ASR administrator stack with the Web UI enabled (`ShouldDeployWebUI=yes`, the default). This creates the Amazon Cognito user pool, the API, and the `asr-api/full-access` scope. You also need:

- AWS CLI v2 authenticated to the ASR administrator account
- `curl` and `jq` for the token and API examples below

Step 1: Create the M2M app client after deployment

Find the user pool id in the administrator stack's `UserPoolId` **Output**, then create a confidential client granted **both** required OAuth 2.0 scopes — `asr-api/api` (Amazon API Gateway requires this scope to admit the request to the API) and `asr-api/full-access` (the backend authorizes Full Access on this scope). A client granted only `asr-api/api` reaches the API but receives a `403` rejection, because the backend treats it as an unprivileged machine token.

```
export AWS_REGION="us-east-1"
export STACK_NAME="<ASR-administrator-stack-name>"
```

```
export USER_POOL_ID="$(aws cloudformation describe-stacks \
  --stack-name "$STACK_NAME" --region "$AWS_REGION" \
  --query "Stacks[0].Outputs[?OutputKey=='UserPoolId'].OutputValue" --output text)"

CREATE_OUT="$(aws cognito-idp create-user-pool-client \
  --user-pool-id "$USER_POOL_ID" --region "$AWS_REGION" \
  --client-name "ASR-M2M-FullAccess" \
  --generate-secret \
  --allowed-o-auth-flows client_credentials \
  --allowed-o-auth-flows-user-pool-client \
  --allowed-o-auth-scopes "asr-api/api" "asr-api/full-access" \
  --output json)"

export M2M_CLIENT_ID="$(echo "$CREATE_OUT" | jq -r '.UserPoolClient.ClientId')"
export CLIENT_SECRET="$(echo "$CREATE_OUT" | jq -r '.UserPoolClient.ClientSecret')"
```

The response contains `UserPoolClient.ClientId` and `UserPoolClient.ClientSecret`. **Cognito shows you the secret value only once.** Cognito requires the secret to be generated when the client is created (`--generate-secret`); there is no command to add a secret to an existing client later.

Step 2: Store the secret in your own secret store

The solution never stores this secret. It is intentionally kept out of AWS CloudFormation outputs and template state. You are responsible for capturing the returned client secret and storing it securely in your own secret store — for example AWS Secrets Manager, AWS Systems Manager Parameter Store (Secure String), or your organization's vault.

- Store the secret immediately; you cannot retrieve it again later.
- Restrict read access to the integration that needs it.
- Anyone who holds this secret can obtain Full Access tokens for ASR, so guard it accordingly.

Step 3: Retrieve an access token (client_credentials flow)

The token endpoint is derived from the user pool's hosted domain. Request a token using HTTP Basic auth (`client_id:client_secret`), the `client_credentials` grant, and both scopes:

```
export DOMAIN_PREFIX="$(aws cognito-idp describe-user-pool \
  --user-pool-id "$USER_POOL_ID" --region "$AWS_REGION" \
```

```
  --query "UserPool.Domain" --output text)"
export TOKEN_URL="https://${DOMAIN_PREFIX}.auth.${AWS_REGION}.amazoncognito.com/oauth2/
token"

export ACCESS_TOKEN="$(curl -s -X POST "$TOKEN_URL" \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -u "${M2M_CLIENT_ID}:${CLIENT_SECRET}" \
  -d "grant_type=client_credentials&scope=asr-api/api asr-api/full-access" \
  | jq -r '.access_token')"
```

Access tokens are valid for **1 hour**, and the `client_credentials` grant does not issue a refresh token. **Cache and reuse a token for the duration of its validity** rather than requesting a new one for every API call. Request a new token shortly before the current one expires — for example, one to two minutes before the 1-hour mark — using the credentials you already hold.

Caching the token keeps your integration within the Cognito token endpoint rate limits and reduces cost. A correctly caching integration needs only about one token request per hour. Requesting a new token for every API call is slower and unnecessarily costly.

Step 4: Call the ASR API

Send the access token as a `Authorization: Bearer <access_token>` header on any ASR API path. API Gateway validates the token and the `asr-api/api` scope before the request reaches the backend, and the backend grants Full Access on the `asr-api/full-access` scope:

```
# Find the API base URL in the administrator stack's APIEndpoint Output, then
# call an endpoint your integration uses.
curl -s -H "Authorization: Bearer ${ACCESS_TOKEN}" \
  "https://<api-id>.execute-api.${AWS_REGION}.amazonaws.com/prod/<endpoint>"
```

You do not need additional network or firewall configuration for programmatic callers. The API's AWS WAF protections already admit authenticated machine clients (non-browser user agents from cloud IP ranges) while blocking abusive traffic. Machine activity is attributed to the client ID in the ASR API logs, and AWS CloudTrail records every token issuance.

Step 5: Rotate or disable the client

Because a secret cannot be added, replaced, or removed on an existing Cognito app client, rotation and disablement operate on the **client** itself:

- **Rotate:** create a second M2M client (repeat Step 1 with a different `--client-name`), migrate your integration to its credentials, confirm it can obtain tokens, then delete the old client.
- **Disable:** delete the client to immediately revoke machine access.

```
aws cognito-idp delete-user-pool-client \
  --user-pool-id "$USER_POOL_ID" \
  --client-id "$M2M_CLIENT_ID" \
  --region "$AWS_REGION"
```

Detect denied machine tokens (optional)

When Amazon CloudWatch alarms are enabled, ASR ships an opt-in alarm (`ASR-Cognito-M2MForbidden`) on the `ASR/M2MForbiddenAuthorization` metric. The API AWS Lambda function emits this metric whenever a machine token is denied for lacking the `asr-api/full-access` scope — a non-zero count signals a misconfigured client or probing. The alarm uses ASR's shared SNS alarm topic, which has no subscriber by default; subscribe a channel to be notified.

Enable Amazon Cognito logging and threat protection

We recommend that you enable native log delivery and Advanced Security on the ASR user pool for visibility into authentication activity.

Amazon Cognito native log delivery

With this option, you can deliver sign-in and token events to Amazon CloudWatch Logs. In the Amazon Cognito console, open your user pool and choose **Logging**. Enable log delivery to a CloudWatch Logs log group. Set the log level to `ERROR` or `INFO` depending on how much detail you need.

Amazon Cognito records token issuance in AWS CloudTrail. To retain and query these audit records in Amazon CloudWatch Logs, configure a management-event CloudTrail trail in the account that owns the user pool. Configure the trail to deliver logs to a CloudWatch Logs log group. For more information, see Monitoring CloudTrail Log Files with Amazon CloudWatch Logs.

Amazon Cognito Advanced Security and threat protection

With this option, you get compromised-credential detection and risk-based adaptive authentication. In the Amazon Cognito console, open your user pool, choose **Advanced Security**, and set the mode to **Enforced**.

Additional cost

If you enable Advanced Security, you incur additional per-authentication charges. For more information about Amazon Cognito pricing, see Amazon Cognito pricing.

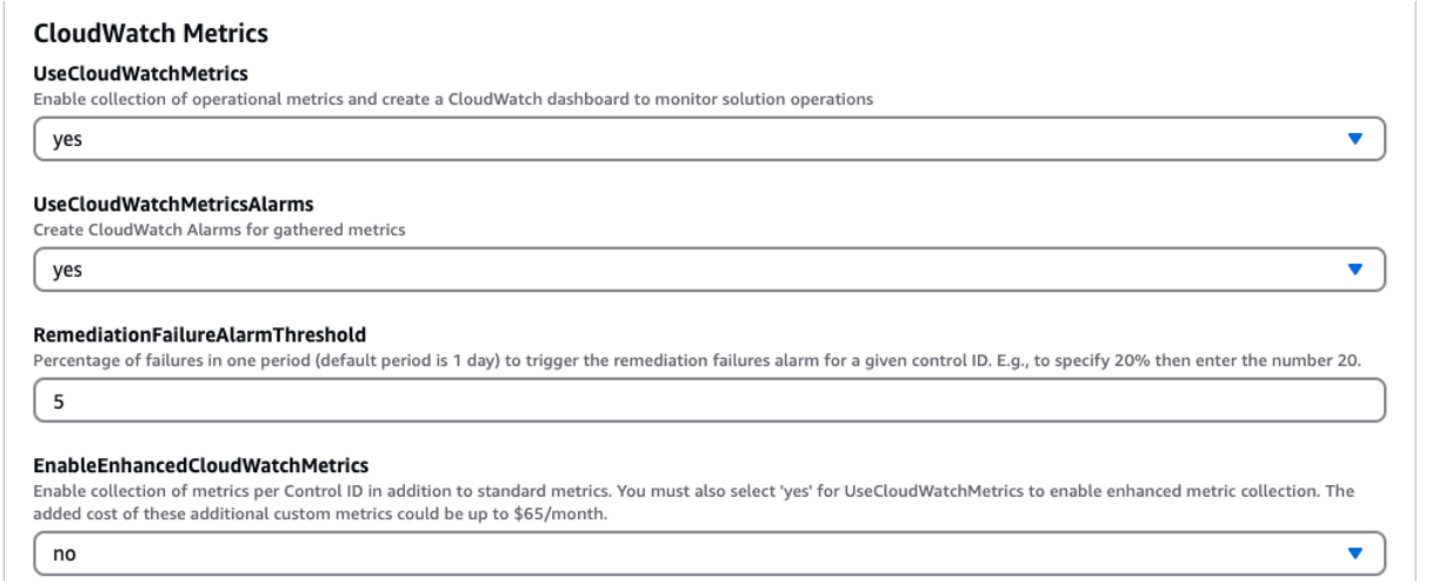
Monitor the solution's operations with an Amazon CloudWatch dashboard

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

The CloudWatch dashboard and alarms monitor the solution’s operations and alert 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:

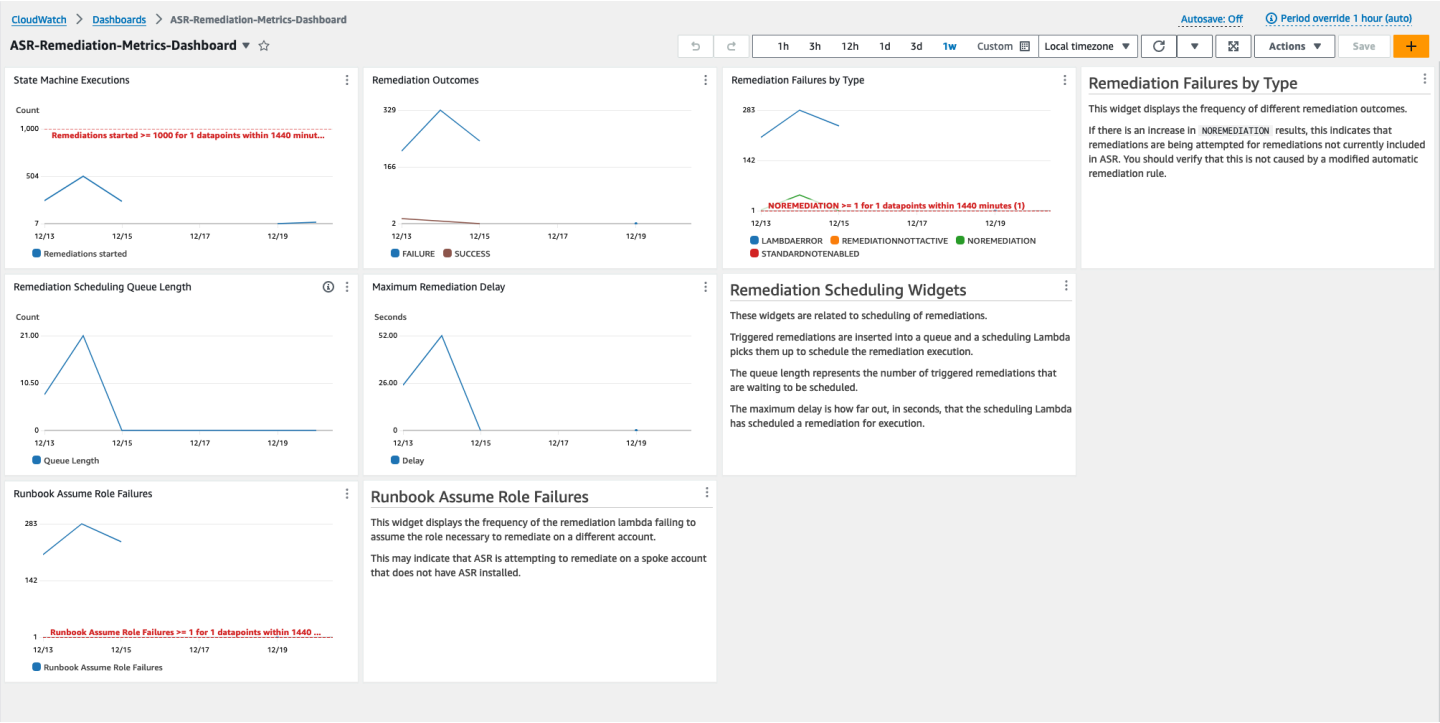
1. **Total Successful Remediations** - Gives you insight into the number of Security Hub findings that have been successfully remediated by the solution.
2. **Remediation Failures** - Shows how many remediations have been failing, both in total and as a percentage, and the failure cause. A high number 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.

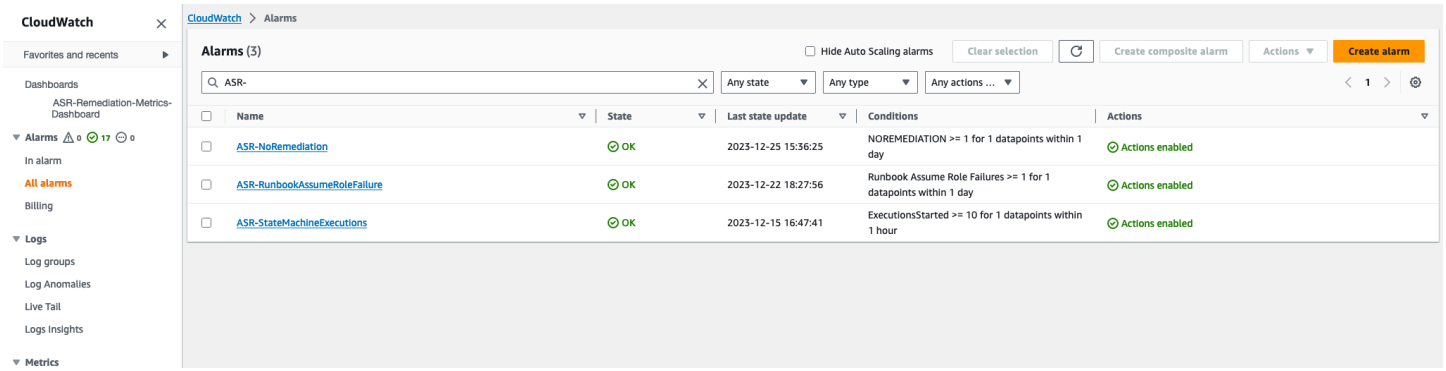
2. Remediation Failures by Type = NOREMEDIATION > 0
    a. Remediations are being attempted for remediations that are not included in ASR. This could indicate an event rule has been modified to include more than the intended remediations.
3. Runbook Assume Role Failures > 0
    a. Remediations are being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

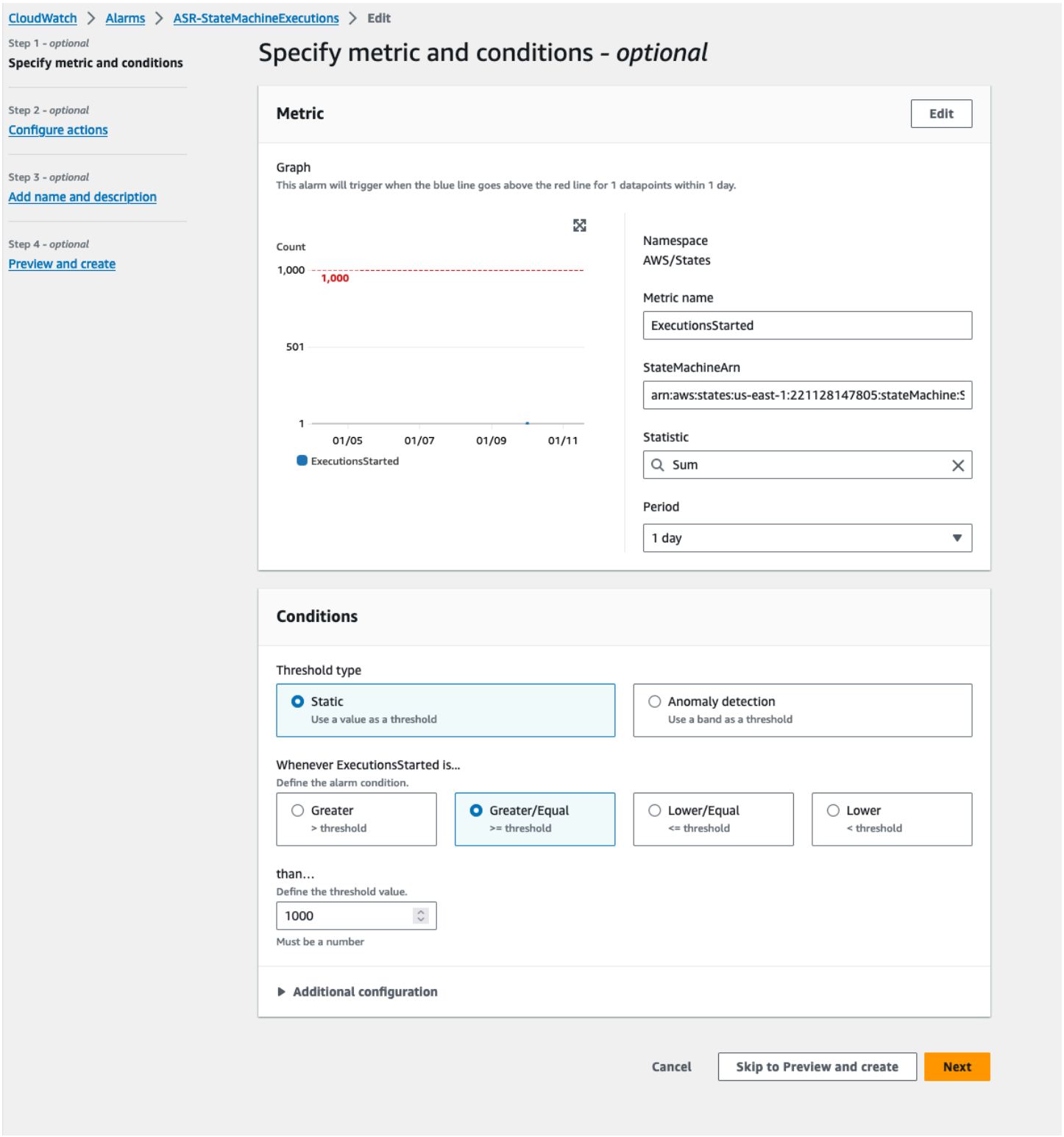


Modifying alarm thresholds

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



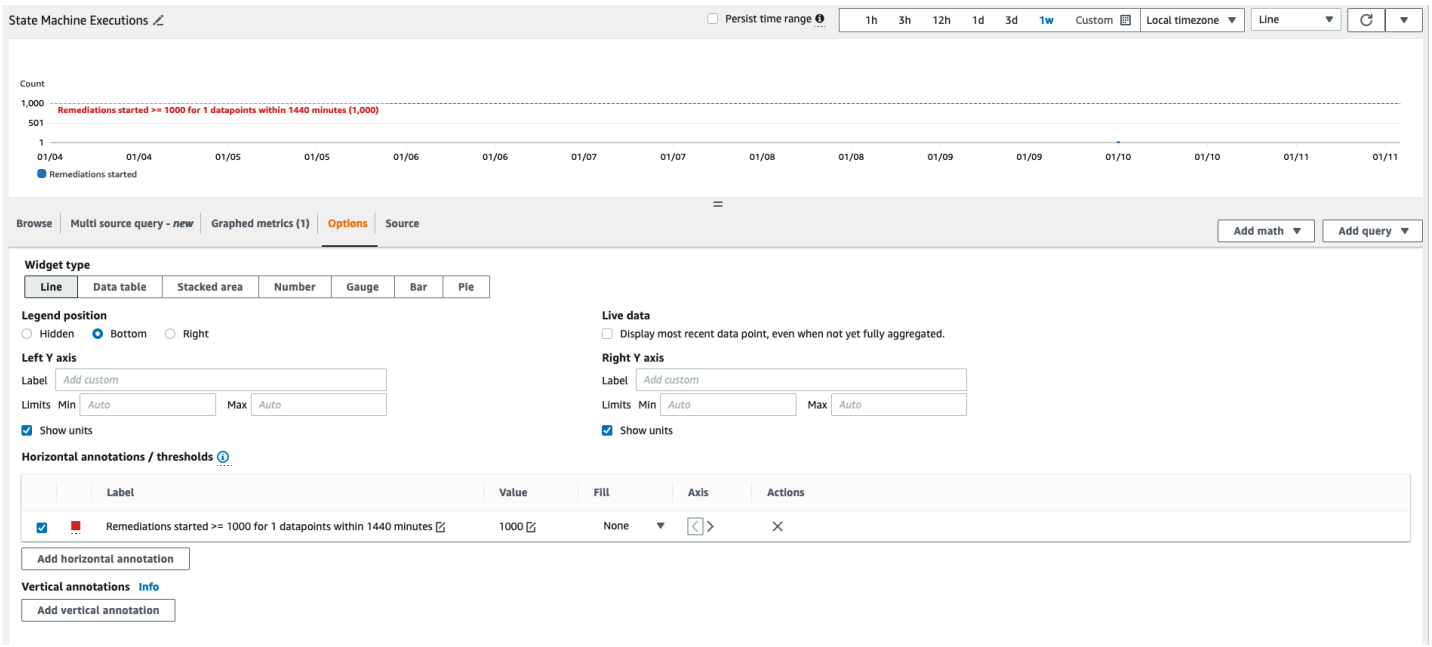
1. Change the threshold to the desired value and save.



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

c. Choose 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

Important

- **Upgrading to v4.0.0 or later:** The solution preserves your existing auto-remediation settings across the upgrade. If you are upgrading from **v2.x**, the solution automatically migrates the auto-remediation settings from your v2 EventBridge rules to the new DynamoDB-backed configuration during the stack update. Controls that you had auto-remediation enabled for in v2 stay enabled in v4, and controls without a v3+ equivalent are skipped and listed in the migration AWS Lambda function's Amazon CloudWatch logs. If you are upgrading from **v3.x**, your auto-remediation settings already reside in the DynamoDB-backed configuration and carry over unchanged, so no migration is required. If the migration encounters a per-control failure or aborts before any controls are written, the `SO0111-ASR-MigrationAutoRemediation` Lambda function publishes an Amazon SNS notification to the existing `SO0111-ASR_Topic`. **To receive that notification, you must have a confirmed subscription on `SO0111-ASR_Topic` before starting the v4 stack update.** Refer to Upgrading from v2.x to v4.0.0 or later.
- **Upgrading to v3.x from v2.x:** Re-enable automated remediation rules manually in the Admin account after the stack update. 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"

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 `Reuse 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.
4. If the Web UI is enabled and you updated parameters such as `TicketGenFunctionName`, invalidate the CloudFront cache to reflect changes immediately:

```
aws cloudfront create-invalidation \
  --distribution-id <distribution-id> \
  --paths "/aws-exports.json"
```

Upgrading from v2.0.x

If you are upgrading from v2.0.x, upgrade to v2.3.0 first. Updating to v2.1.0–2.1.1 fails in CloudFormation.

Upgrading from v2.1.4 or earlier

If you are upgrading from v2.1.4 or earlier, **you must upgrade to v2.3.0 before upgrading to any version higher than v2.3.0.** Otherwise, the stack update operation will fail. Alternatively, you can delete and re-deploy the solution's stacks rather than performing a stack update.

Upgrading from v2.x to v4.0.0 or later

Starting in v4.0.0, the solution preserves your auto-remediation settings across the v2 to v4 upgrade. In v2, per-control Amazon EventBridge rules stored the auto-remediation state. In v3 and later, the solution stores it in the Remediation Configuration Amazon DynamoDB table in the admin account. During the v4 stack update, a custom resource scans your existing v2

_AutoTrigger rules, translates each rule's standard-specific control ID to the corresponding security control ID, and writes the previously enabled controls to the DynamoDB table.

The migration covers all five v2 playbooks:

- SC (AWS Security Hub Service-managed security controls) — control IDs are written to DynamoDB unchanged.
- AFSBP (AWS Foundational Security Best Practices) — control IDs are written to DynamoDB unchanged.
- NIST80053R5 (NIST 800-53 Revision 5) — control IDs are written to DynamoDB unchanged.
- PCI (PCI DSS v3.2.1) — the leading PCI. prefix is removed (for example, PCI.S3.5 becomes S3.5).
- CIS (CIS AWS Foundations Benchmark v1.2.0, v1.4.0, and v3.0.0) — each CIS control ID is mapped to the security control its v2 SSM remediation document targeted.

Migration outcomes:

- Controls migrated successfully need no action — they are auto-remediation enabled in v4 the same way they were in v2.
- Controls that the migration skips fall into two categories: CIS rules without an ASR remediation in v2, and controls v3+ does not ship. Skipped controls are listed in the S00111-ASR-MigrationAutoRemediation AWS Lambda function's Amazon CloudWatch logs.
- If the migration encounters a per-control failure (for example, a transient throttling error during the DynamoDB update) or aborts before any controls are written, S00111-ASR-MigrationAutoRemediation publishes an Amazon SNS notification to the existing S00111-ASR_Topic topic. The notification body lists the affected control IDs and is prefixed with [ASR v2 # v3/v4 migration].
- The custom resource only runs during the initial v3/v4 stack update. Subsequent stack updates do not re-run migration.

> **Note**
>
> **Subscribe to the SNS topic before upgrading.** Migration failure notifications publish to the solution's existing S00111-ASR_Topic Amazon SNS topic. If you do not have a confirmed subscription on that topic before the upgrade, the failure notification does not reach you.

In that case, you only see the failure in the `SO0111-ASR-MigrationAutoRemediation` AWS Lambda function's Amazon CloudWatch logs. To subscribe an endpoint, retrieve the topic ARN from AWS Systems Manager Parameter Store at `/Solutions/SO0111/SNS_Topic_ARN` and add an SNS subscription (email, SQS, AWS Lambda function, or any other supported protocol) before starting the v4 stack update. Confirm any email subscriptions through the AWS confirmation email so the topic is ready to deliver before the migration runs.

If migration is unable to write a control you want auto-remediated in v4, enable it manually in the Remediation Configuration DynamoDB table. Refer to Enable fully-automated remediations.

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 managed key.

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-e098549f5259`.

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, 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 policy 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. In the Admin account, navigate to the AWS CloudFormation console.
2. Locate the Admin stack and view its **Outputs** tab.
3. Copy the value of the `RemediationConfigurationDynamoDBTable` output.
4. Navigate to the DynamoDB console and open the Remediation Configuration table.
5. Select **Explore Table Items**.

6. Under **Scan or query items**, select **Query**.
7. Enter the control ID (for example, `Lambda.1`) in the **Partition key: controlId** field and choose **Run**.
8. Select the returned item, then choose **Actions > Edit item**.
9. Change the `automatedRemediationEnabled` attribute value to **False**.
10. Choose **Save and Close**.

Scenario 2: Disable automatic remediation for all controls

1. Follow steps 1-5 from Scenario 1 to access the Remediation Configuration table items.
2. Under **Scan or query items**, select **Scan** to view all controls.
3. For each control with `automatedRemediationEnabled` set to **True**, select the item and choose **Actions > Edit item**.
4. Change the `automatedRemediationEnabled` attribute value to **False** and choose **Save and Close**.
5. Repeat for all controls you wish to disable.

Scenario 3: Disable manual remediation for an account

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

Contact AWS Support

If you have AWS Business Support+, AWS Enterprise Support, or Unified Operations, you can use the AWS 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 **Automated Security Response on AWS**.
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, including the name of this solution and the version you are using, such as this example: **Automated Security Response on AWS vX.Y.Z**.
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 playbooks, FSBP playbooks, or both. 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 from the Security Hub member accounts.
4. Remove the spoke `AFSBPMemberStack.template` template from 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. Choose 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.

Note

The solution's Amazon DynamoDB tables are retained when you remove the admin stack, so your remediation history and configuration are preserved. These tables are the Findings, Remediation History, Remediation Configuration, Resource Filters, Notification

Configuration, and Notification Batches tables. Each has deletion protection enabled. To remove a table, first disable its deletion protection, then delete the table.
Any AWS Secrets Manager secrets you created for Slack, JIRA, or ServiceNow channels are also retained. Delete them manually if you no longer need them.

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) 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) 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) 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 can initiate fully automated remediations only for controls that are enabled in the Remediation Configuration Amazon DynamoDB table. This table is deployed to the admin account.
- For simplicity, deploy 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) 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) 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".

Which controls have fully-automated remediation enabled:

The solution offers two ways to enable or disable fully-automated remediation. Both write to the same Remediation Configuration Amazon DynamoDB table in the admin account, so they stay in sync.

- **Web UI (recommended when deployed).** Manage fully-automated remediation from the **Controls** page. Admins and delegated admins can enable or disable automated remediation per control, and apply reusable resource filters to scope which findings are acted on. To turn off automated remediation for all controls at once — for example, during an incident, choose **Disable All Automated Remediations**. This button is unavailable to account operators (read-only) and when automated remediation is already off for all controls. For details, see Manage automated remediation.
- **Legacy configuration (no Web UI).** If you did not deploy the Web UI, enable or disable fully-automated remediation by editing the items in the Remediation Configuration DynamoDB table directly, and scope which findings are remediated with the AWS Systems Manager filter parameters. For the steps, see Enable fully-automated remediations. To quickly disable automated remediation during an incident without the Web UI, see the disable scenarios in Troubleshooting.

Access to the solution's Web UI:

- When the Admin stack is deployed, you receive an email with temporary credentials to sign in to the Web UI using the email address you provided during deployment.
- Using the **Invite Users** page, administrators and delegated administrators can invite additional users to access the Web UI and delegate access to the solution.
- Using the **View Users** page, administrators and delegated administrators can view and manage existing users.
- To learn more about permissions and how to use the solution's Web UI, see the the section called "Web UI".

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"
 }
}
```

Configure notifications

In addition to the solution-managed Amazon SNS topics described in Example SNS notifications, you can create notification configurations from the Web UI. Each configuration controls which events are delivered, where they are delivered, and what each message contains. The solution stores configurations in the admin account and applies them to findings and remediations aggregated to that account and Region.

Web UI required

You create and manage notification configurations from the Web UI. You must enable the Web UI (deploy the Web UI components when you launch or update the admin stack) to create and manage notification configurations.

Notification channels

Each notification configuration delivers events through one or more channels. The solution supports the following channel types:

Channel	Description
Email	Sends email to the recipients you select. Recipients can be chosen by role (primary account contact, security contact, operations contact, account operators) or entered as custom email addresses. Custom email recipients must confirm their subscription before they receive messages.
Slack	Posts messages to a Slack channel. Requires the Slack channel ID and an AWS Secrets Manager secret holding the Slack credentials.
JIRA	Creates a JIRA issue. Requires the JIRA instance URL, project key, issue type, and a Secrets Manager secret holding the JIRA credentials.
ServiceNow	Creates a record in a ServiceNow table. Requires the ServiceNow instance URL, table name, and a Secrets Manager secret holding the ServiceNow API key.

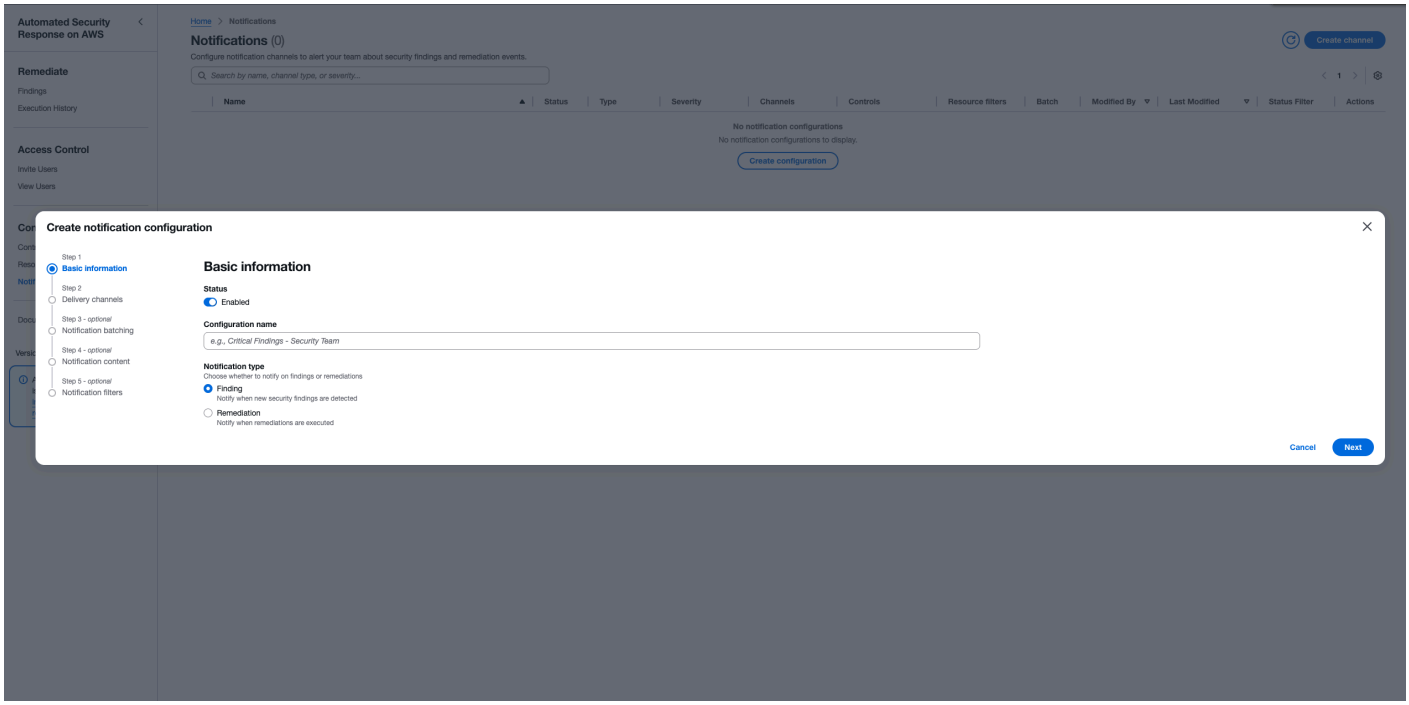
Channel	Description
Amazon SNS	Publishes to an Amazon SNS topic in the admin account and Region.

Important

The solution reads credentials for the Slack, JIRA, and ServiceNow channels from AWS Secrets Manager. The secret name must start with `asr/notifications/` (for example, `arn:aws:secretsmanager:us-east-1:111111111111:secret:asr/notifications/servicenow-creds`). The instance URL for the JIRA and ServiceNow channels must use HTTPS.

Create a notification configuration

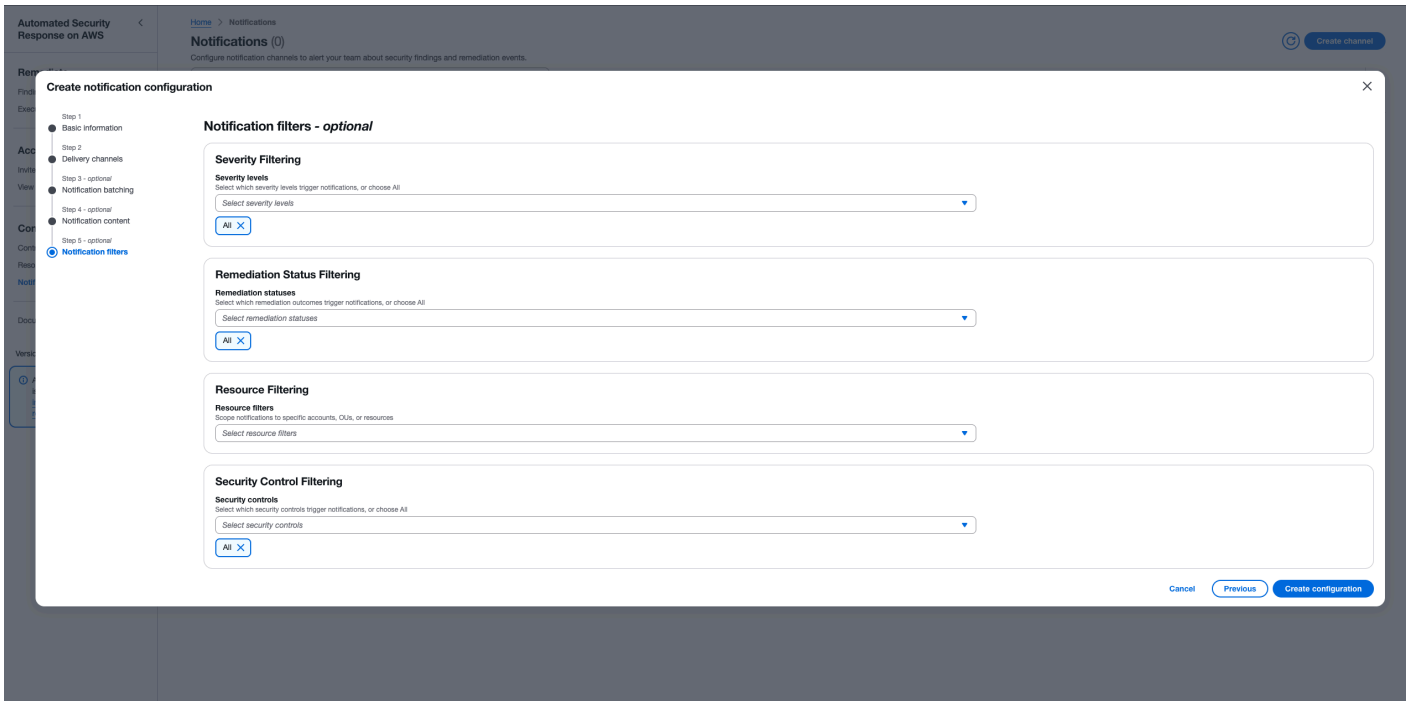
1. Open the Web UI and go to the **Notifications** page.
2. Choose **Create notification configuration**.



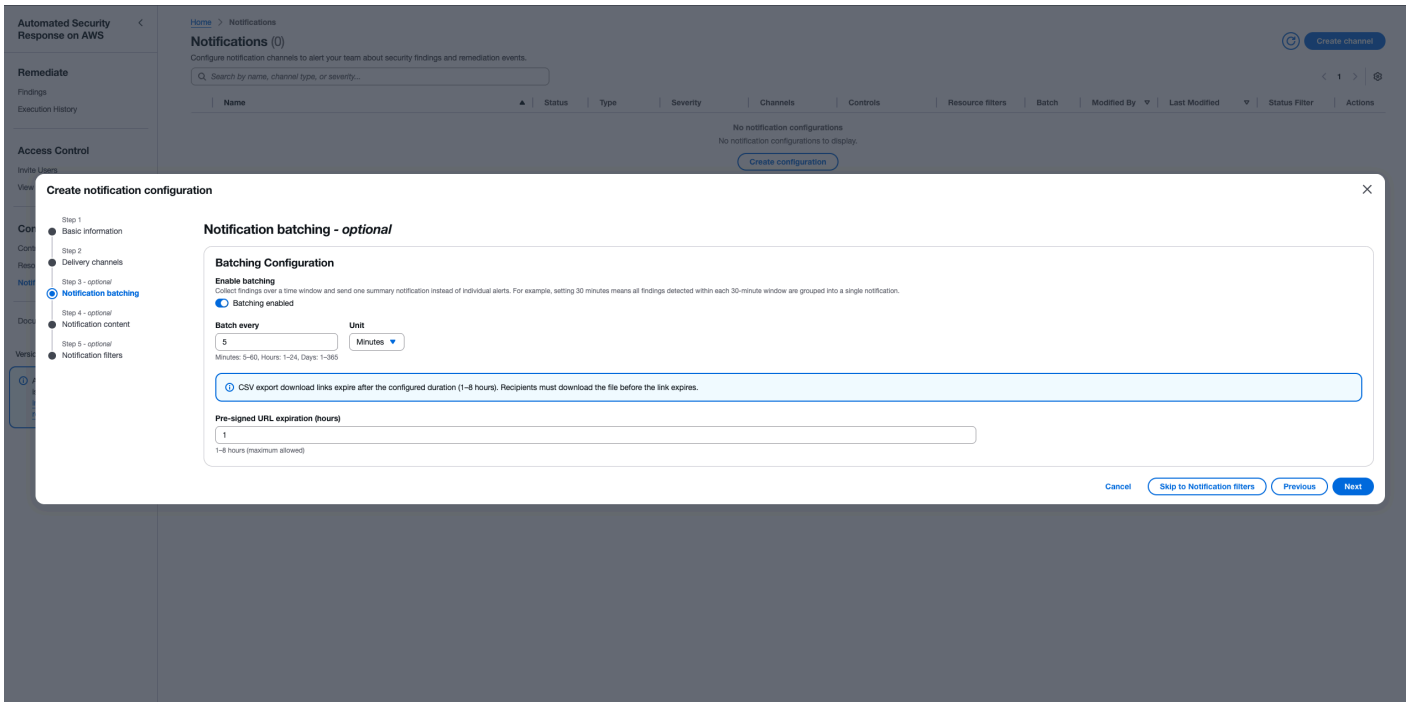
3. **Basic information** – Enter a configuration name and choose the notification type:
   a. **Finding** – Notify when new findings are detected.
   b. **Remediation** – Notify when remediations are executed.

4. **Delivery channels** – Select one or more channels and provide their configuration. At least one channel must be enabled.

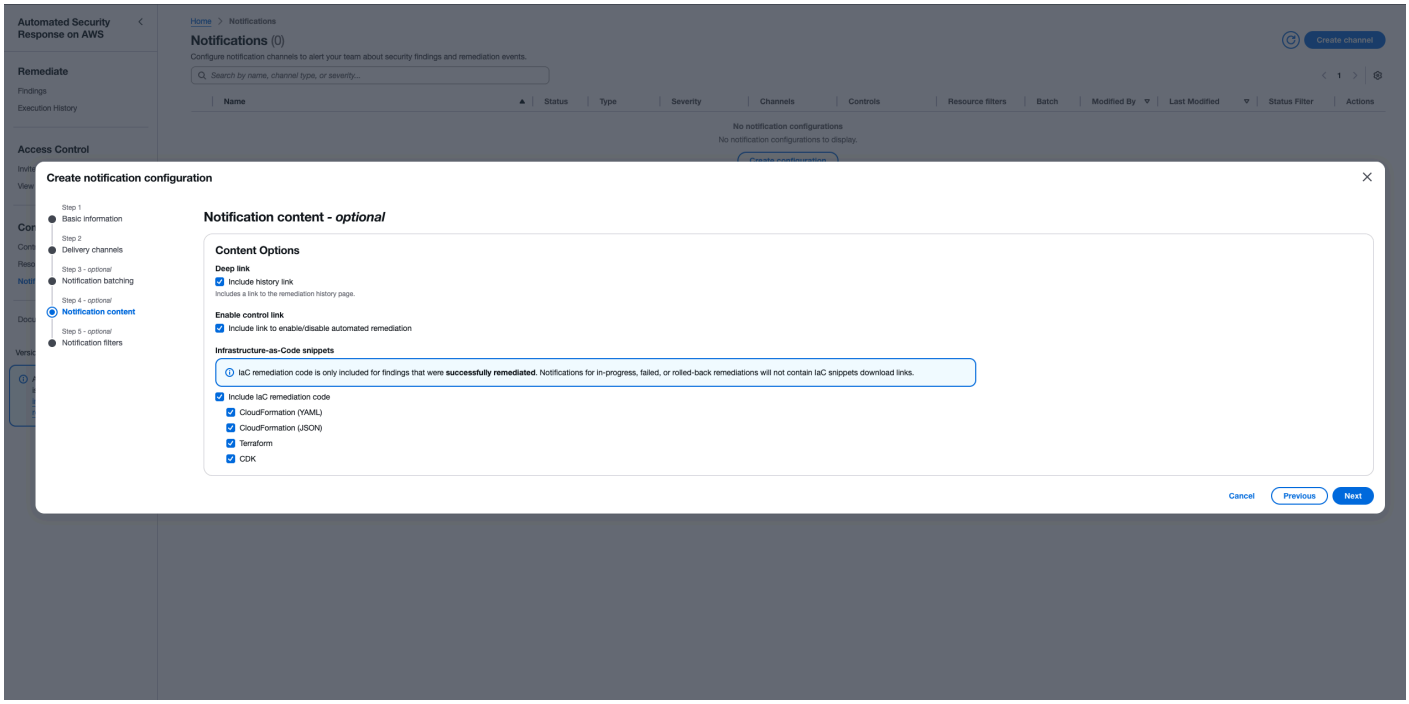
5. (Optional) **Filters** – Narrow the events the configuration matches by severity, remediation status (remediation type only), control IDs, or resource filters. Leaving a filter set to **All** applies no restriction on that dimension.



6. (Optional) **Batching** – Enable a batch window to group multiple events into a single notification. The batch window duration must be within these ranges: 5–60 minutes, 1–24 hours, 1–365 days.



7. **Content options** – Choose what each message includes. See [Notification content options].



8. Choose **Submit**.

Notification content options

Content options control what each notification message includes. The following options each add a link or snippet to the message:

Option	Description
`includeManualRemediationLink`	Adds a link to the finding's page in the Web UI, where an operator can run the remediation manually.
`includeEnableAutomationLink`	Adds a link to the Web UI page for enabling fully-automated remediation for the control.
`includeIaCSnippet`	Adds an infrastructure-as-code snippet for the remediation in the formats you select (CloudFormation YAML, CloudFormation JSON, Terraform, or CDK). Applies to remediation-type notifications.
`includeRemediationDeadline`	Adds a remediation due-by date to each notification for a matching finding. Applies to finding-type notifications. When enabled, set `remediationDeadlineDays` (1–90). See Remediation deadlines and deadline enforcement.

Links require the Web UI

The `includeManualRemediationLink` and `includeEnableAutomationLink` options add links to Web UI pages. The Web UI must be enabled for these links to resolve. If the Web UI is not deployed, do not enable these options.

Remediation deadlines and deadline enforcement

A finding-type notification configuration can attach a remediation deadline to each finding it matches, and — optionally — enforce that deadline by automatically remediating overdue findings.

Remediation deadlines

Enable the `includeRemediationDeadline` content option on a finding-type configuration and set `remediationDeadlineDays` to a value from 1 to 90. For each finding the configuration matches, the solution computes a remediation due-by date (the finding's creation time plus the configured number of days) and includes it in the notification.

By themselves, deadlines are **informational**. The due-by date appears in the notification, but nothing happens when it passes — the solution does not act on an overdue finding unless you also enable enforcement.

Deadline enforcement

Warning

When you enable the `enforceDeadline` option on a finding-type configuration, remediation deadlines stop being advisory and become **enforcing**. The solution automatically triggers the Orchestrator to remediate any overdue, unresolved finding that matches the configuration — **with no human intervention**. This applies whether or not the finding's control has automated remediation enabled. Enable enforcement only for configurations whose matching findings you are comfortable having the solution remediate automatically.

24-hour minimum grace period

The solution never remediates a finding less than **24 hours** after it becomes eligible for enforcement — even for a pre-existing backlog whose creation-time deadline is already in the past. This 24-hour floor is fixed and non-configurable, and is separate from (and additional to) the 1–90 day `remediationDeadlineDays` range.
As a result, turning on enforcement for an existing backlog does **not** trigger a mass remediation the instant you enable it. Every matching finding gets at least 24 hours of notice before the solution can remediate it.

Prerequisites and scope

- **Finding-type only.** Enforcement is supported only on finding-type notification configurations, not remediation-type configurations.

- **Deadline required.** The configuration must have `includeRemediationDeadline` enabled and `remediationDeadlineDays` set to a value from 1 to 90.
- **Web UI required.** Notification configurations are created and managed from the Web UI, so the Web UI must be deployed.
- **Config-boundary scope.** Enforcement applies to findings that match the configuration's control, severity, and resource filters — the notification-config boundary, not the control boundary. It does not depend on whether the matching control has automated remediation enabled. In practice, only findings that were not already auto-remediated ever reach enforcement.

ServiceNow custom field mappings

Custom field mappings let you add key-value pairs to the ServiceNow record created for a finding. The value can include template variables that the solution substitutes from the finding at send time. The key is the ServiceNow column name on the target table.

The following template variables are supported in the value:

- `${FINDING_ID}`
- `${CONTROL_ID}`
- `${SEVERITY}`
- `${ACCOUNT_ID}`
- `${REGION}`
- `${RESOURCE_ARN}`
- `${CONFIG_NAME}`

The solution enforces the following constraints on custom field mappings:

- Up to 20 mappings per channel.
- Key length up to 64 characters; resolved value length up to 256 characters.
- Do not use the following solution-managed fields as keys: `short_description`, `description`, `urgency`, `impact`, `work_notes`.

Important

The solution writes finding data to ServiceNow as-is and does not HTML-encode or sanitize the values it sends. Map custom fields only to standard data fields (for example, plain String fields). Do **not** map them to:

- HTML fields or journal/journal-input fields, which can render HTML, or
- Script fields (`script`, `script_plain`), which can be evaluated by server-side code.

A plain String field displays values as text. HTML and journal fields can render markup. Finding data, such as a resource identifier, might contain characters that your ServiceNow instance renders as markup. Whether stored values are rendered or escaped is governed by your ServiceNow instance's security hardening settings (for example, HTML escaping and the HTML sanitizer), which you own and should keep enabled. Prefer customer-defined custom columns (typically prefixed u_ or x_) over out-of-box fields, and confirm the target field type before mapping.

For more information about ServiceNow security hardening, see the following resources:

- ServiceNow input validation (instance security hardening)
- Render journal field entries as HTML (default High Security escaping behavior)

Manage automated remediation

When the Web UI is enabled, you manage automated remediation from the **Controls** page instead of editing the Remediation Configuration Amazon DynamoDB table or the AWS Systems Manager parameters directly. From this page you enable automated remediation for individual controls and scope which findings the solution acts on by applying reusable resource filters.

Web UI required

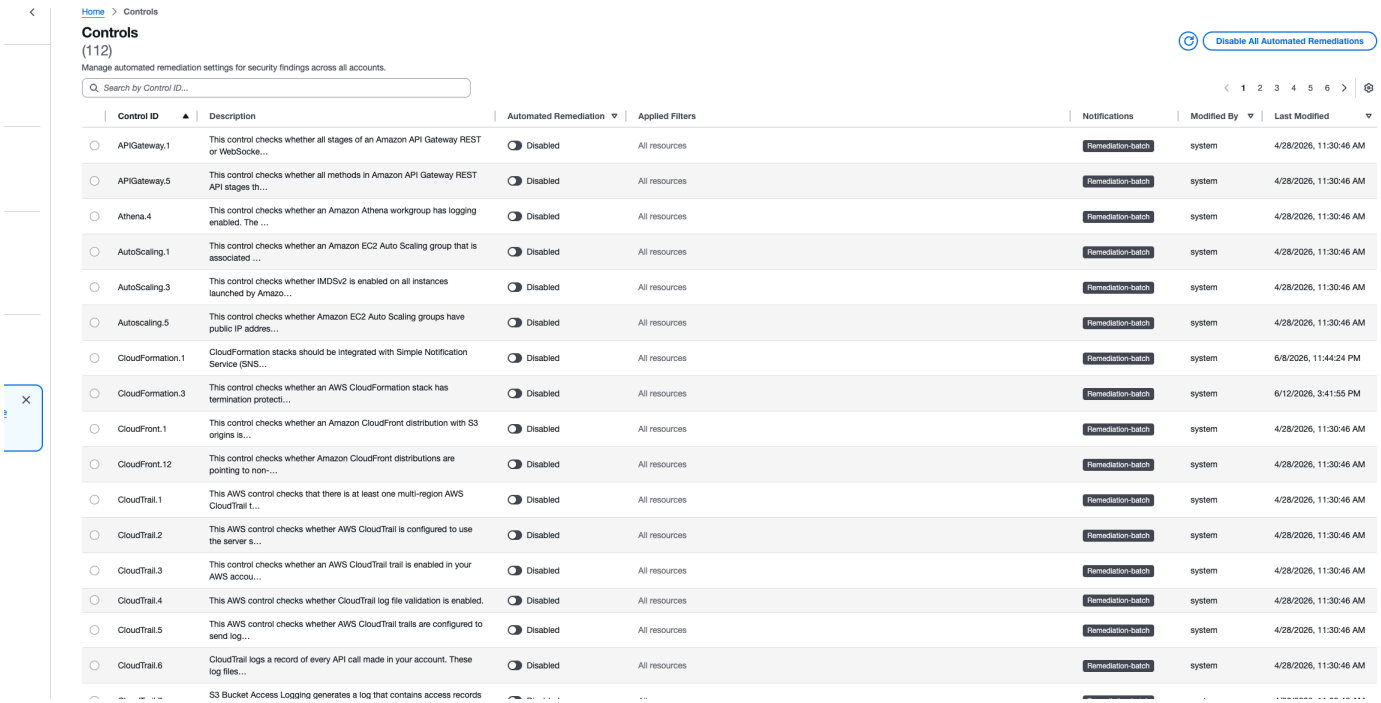
The Controls page and resource filters require the Web UI. If you did not deploy the Web UI, configure automated remediation through the Remediation Configuration DynamoDB table and the Systems Manager filter parameters, as described in Enable fully-automated remediations.

Access to the Controls page follows the same roles as the rest of the Web UI. Admins and delegated admins can enable or disable automated remediation, and create, edit, and apply filters. Account operators have read-only access to the Controls and Filters pages.

Enable automated remediation for a control

The Controls page lists every security control that the solution supports, with its current automated remediation status, description, and applied filters. You can search and sort the table by control ID or filter name.

1. Open the Web UI and go to the **Controls** page.



2. Set the automated remediation status for one or more controls.
3. (Optional) Apply one or more resource filters to a control, and choose a filter mode (include or exclude).
4. Choose **Save** to persist your changes. The solution validates and applies all pending changes together.

> **Deadline enforcement is a separate path to automated remediation**
>
> Enabling automated remediation on this page is the primary path: when a matching finding arrives, the solution remediates it only if the control is enabled here (subject to any applied resource filters). A second, independent path exists — **deadline enforcement**. If

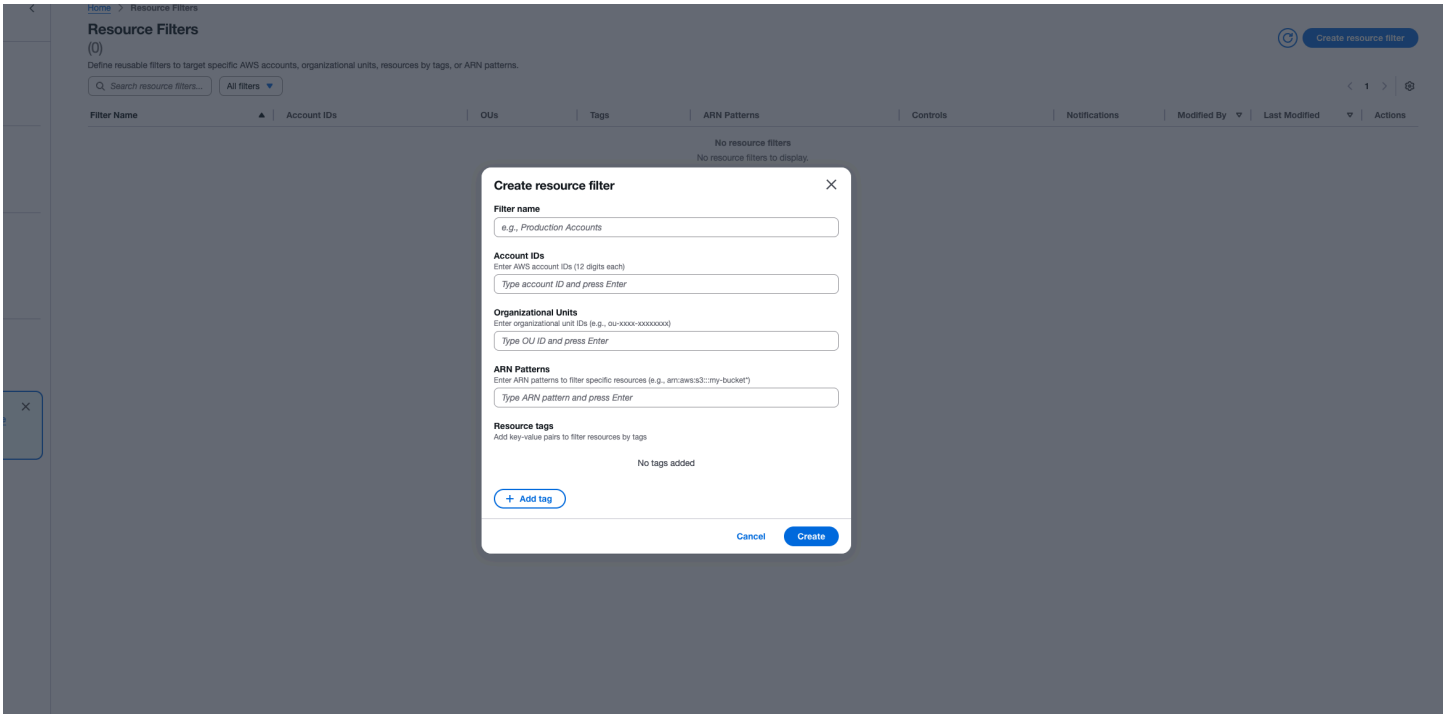
a finding-type notification configuration has `enforceDeadline` turned on, the solution automatically remediates matching overdue findings **regardless of whether the control is enabled on this page**. If you disable a control here to prevent automated remediation, also confirm that no enforcing notification configuration matches those findings. For details, see Deadline enforcement.

Create and apply resource filters

A resource filter is a reusable definition that scopes which findings the solution automatically remediates. Each filter can combine the following criteria:

- **Account IDs** – One or more 12-digit AWS account IDs.
- **Organizational units** – One or more AWS Organizations OU identifiers.
- **ARN patterns** – One or more resource ARN patterns. Patterns support the wildcard in any ARN component (for example, `arn:aws:` matches any resource in account 111111111111).
- **Resource tags** – One or more tag key-value pairs. The solution ignores tag criteria for resource types that cannot be tagged, such as IAM users and AWS accounts.

To create a filter, open the **Filters** page and choose **Create filter**, then enter a unique name and the criteria to match.



You apply a filter to a single control from the Controls page, or to every control at once with the **Apply to all controls** action on the Filters page. Removing a filter works the same way. Deleting a filter removes it from every control that references it.

How filters are evaluated

When a finding arrives, the pre-processor decides whether to automatically remediate it based on the control's status and the filters applied to that control:

- If automated remediation is disabled for the control, the solution does not remediate the finding.
- If automated remediation is enabled and the control has no filters, the solution remediates the finding.
- **Include mode** – The solution remediates the finding only when it matches **all** filters applied to the control (logical AND).
- **Exclude mode** – The solution does not remediate the finding when it matches **any** filter applied to the control.

A finding matches a filter when the finding's account ID, organizational unit, resource ARN, or resource tags match the corresponding criteria in the filter.

Tutorial

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 using the Remediation Configuration DynamoDB table. 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**
>
> Make sure you choose the option to "Include global resources (such as AWS IAM resources)." If you do not choose this option when enabling AWS Config, you do not see findings related to global resources (such as 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 might take some time for findings to be generated with the new feature. You can proceed with the tutorial, but you are unable 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 preceding example, this tutorial uses the manual invitation method. For a set of production accounts, we recommend managing Security Hub delegated administration 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 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 creates an example resource with an insecure configuration to demonstrate a remediation. The example control is Lambda.1: Lambda function policies should prohibit public access.

> **Important**
>
> You will intentionally create a resource with an insecure configuration. Review the nature of the control and evaluate the risk of creating such a resource in your environment. Be aware of any tooling your organization might have for detecting and reporting such resources and request an exception if appropriate. If the example control is inappropriate for your environment, 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

You should receive a CREATE_COMPLETE status in the AWS CloudFormation console.

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

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

You should receive a `CREATE_COMPLETE` status for each stack instance in the AWS CloudFormation StackSets console.

Deploy the member roles stack

automated-security-response-member-roles.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 should receive a `CREATE_COMPLETE` status for each stack instance in the AWS CloudFormation StackSets console.

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

Subscribe to the SNS topic

Remediation Updates

Topic - SO0111-ASR_Topic

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

Alarms

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

> **Important**
>
> This example requires the use of the Security Hub CSPM console. The Security Hub (non-CSPM) console does not currently support manual remediations via custom action. To remediate findings without using the Security Hub CSPM console, see the Remediate using the Web UI section.

In the admin account, navigate to the Security Hub CSPM 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
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

Remediate using the Web UI

Alternatively, you can use the solution's Web UI to remediate AWS Security Hub findings and view past remediations.

> **Note**
>
> You must set the `ShouldDeployWebUI` parameter to "yes" when deploying the Admin stack in order to use the solution's Web UI.

Log in to the Web UI

After deploying the solution, you will receive an email with temporary credentials and a link to the solution's Web UI from no-reply@verificationemail.com. This will be sent to the email address you provided when deploying the Admin stack.

Locate the email, copy the temporary credentials, and choose the Web UI link. This link will take you directly to the sign-in page, where you will enter your temporary credentials and set a new password.

Locate the Lambda.1 finding

Once you log in, you will be presented with the **Findings page**. This page displays all Security Hub findings in your Security Hub administrator account that are supported for remediation, including findings for member accounts onboarded with AWS Security Hub.

On the **Findings page**, use the search bar to filter on **Resource ID** by entering the ARN of the Lambda function you created as part of this tutorial and performing a search using the "=" operator. This will display all AWS Security Hub findings supported by the solution for the Lambda function you created.

To find the `Lambda.1` finding generated in this tutorial, apply another filter on **Finding Type**. Choose the **search bar**, select **Finding Type**, and select the **"=" operator**. If consolidated control findings is enabled in your environment, enter `security-control/Lambda.1`. Otherwise, choose a security standard that supports the Lambda.1 control and enter the Generator ID; for example `aws-foundational-security-best-practices/v/1.0.0/Lambda.1`.

After applying the **Resource ID** and **Finding Type** filters, you will see only the Lambda.1 finding generated by AWS Security Hub for your test resource listed in the table.

> **Note**
>
> It may take some time for AWS Security Hub to generate the Lambda.1 finding for the resource you created. If you do not see the finding after applying both filters, wait 5-10 minutes and search for the finding again.

Initiate the remediation

Select the finding you located in the previous step, then choose **Actions > Remediate**. This will begin a remediation for the finding you selected.

You can view the progress of this remediation on the **Execution History** page. After waiting a few minutes, refresh the **Execution History** page by choosing the **refresh** icon on the top right, and you should see that the **Status** has changed from `In progress` to `Success`.

Confirm that the remediation resolved the finding

When the finding is marked as `Resolved` by AWS Security Hub, it will automatically be removed from the **Findings** page in the Web UI.

To verify that the remediation resolved the finding, navigate to the Lambda console in the member account and confirm that the public access has been revoked.

> **Note**
>
> Some findings may still appear on the **Findings** page even with a **Remediation Status** of `Success`. This is because AWS Security Hub takes up to 24 hours to mark a finding as resolved after the resource has been updated. You can **suppress** findings you no longer want to see on the **Findings** page by selecting the finding and choosing **Actions > Suppress**.

Batch remediate a backlog of findings using the API

The Web UI and the AWS Security Hub CSPM custom action are convenient for remediating a small number of findings interactively. To work through a large backlog of existing findings programmatically — for example, immediately after deploying the solution — call the solution's REST API directly.

The same REST API that backs the Web UI (named **AutomatedSecurityResponseApi** in the API Gateway console) exposes a `POST /findings/action` endpoint that accepts a batch of findings to remediate in a single request. This lets you script remediation of a backlog rather than selecting findings by hand.

> **Note**
>
> This endpoint is available only when the Web UI is enabled (deploy the Admin stack with `ShouldDeployWebUI` set to "yes"). You can find the API's invoke URL for the `prod` stage in the API Gateway console for the **AutomatedSecurityResponseApi** API.

Authenticate

All API requests require an Amazon Cognito access token issued for the solution's user pool, passed in the `Authorization` header as a bearer token. The token must carry the `asr-api/api` scope. Use a token for one of the solution's roles (Admin, Delegated Admin, or Account Operator); the same role-based authorization that applies in the Web UI applies to the API. For unattended automation, obtain a token using the machine-to-machine (client credentials) OAuth flow rather than an interactive user sign-in.

Collect the findings to remediate

Use `POST /findings` to search and page through the findings you want to remediate, and collect the `findingId` of each. The findings returned are scoped to the accounts your role is authorized for, so an Account Operator receives only findings for their assigned accounts.

Call the findings action endpoint

Send the collected finding IDs to `POST /findings/action` with an `actionType` of `Remediate`. Each finding is routed through the Orchestrator exactly as it would be from the Web UI.

```
POST /findings/action
Authorization: Bearer <access-token>
Content-Type: application/json

{
  "actionType": "Remediate",
  "findingIds": [
```

```
    "arn:aws:securityhub:us-east-1:111111111111:security-control/Lambda.1/finding/...",
    "arn:aws:securityhub:us-east-1:111111111111:security-control/S3.5/finding/..."
  ]
}
```

`actionType` also accepts `RemediateAndGenerateTicket` (remediate and open a ticket in your configured ticketing channel), and `Suppress` / `Unsuppress` to change a finding's Security Hub workflow status. A `Remediate` or `RemediateAndGenerateTicket` request returns HTTP `202` with a body of `{"status": "IN_PROGRESS"}`; `Suppress` and `Unsuppress` return HTTP `200`.

> **Important**
>
> Send at most **50 finding IDs per request**. To remediate a larger backlog, split the finding IDs into batches of 50 or fewer and issue one request per batch. The API is also protected by rate-based rules (see the AWS WAF configuration in API Gateway Security Policy), so pace successive batches rather than sending them all at once.

Monitor progress

Remediations initiated through the API run asynchronously, just like those started from the Web UI. Track their progress on the **Execution History** page of the Web UI, through the solution's Amazon SNS notifications, or by querying `POST /remediations`.

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 remediation runbook output. For Orchestrator Step Functions logs, refer to the "**SO0111-ASR-Orchestrator**" log group.

Enable fully automated remediations

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

Important

Before enabling fully automated remediations, confirm the solution is configured in the accounts and Regions where you want the solution to make automated changes. To narrow the scope of the solution's automated remediations, see filtering fully automated remediations.

Note

Findings that are suppressed in AWS Security Hub are not automatically remediated, even when fully automated remediation is enabled for the corresponding control. A finding is suppressed when its workflow status is SUPPRESSED. The solution skips suppressed findings during automatic initiation, so suppressing a finding in Security Hub is an effective way to exclude specific findings from automated remediation while leaving the control

enabled. Suppressed findings can still be remediated on demand from the Web UI or the AWS Security Hub CSPM custom action.

Automatic remediation retry limit

When a fully automated remediation fails, AWS Security Hub re-imports the still-failing finding, which would otherwise cause the solution to trigger the remediation again indefinitely. To prevent this, the solution retries a failed finding a maximum of 3 times, waiting at least 15 minutes between attempts. After the third attempt, the solution stops automatically retrying that finding and emits the `RemediationRetryCapReached` Amazon CloudWatch metric. The finding remains visible in the solution and can still be remediated on demand from the Web UI or the AWS Security Hub CSPM custom action. This limit applies per finding and does not affect other findings or controls.

Example: Enable fully automated remediations for Lambda.1

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

Locate the Remediation Configuration DynamoDB Table

In the Admin account, view the `Outputs` for the Admin stack in the CloudFormation console. You will see an output titled `RemediationConfigurationDynamoDBTable`.

This is the name of the Remediation Configuration DynamoDB table, which controls automated remediation configurations for the solution. Copy the value of this output and locate the corresponding DynamoDB table in the DynamoDB console.

> **v2 to v4 migration**
>
> If you upgraded from v2.x to v4.0.0 or later, the solution automatically populates this table with the controls that had auto-remediation enabled in v2. Items written by the migration have `modifiedBy` set to `v2-migration`, so you can distinguish migrated entries from manual edits. To verify that the migration succeeded for the controls you expect, query the table with `automatedRemediationEnabled = true` and review the result. Any controls the migration skipped (CIS rules without an ASR remediation in v2, or controls v3+ does not ship) need to be enabled manually using the steps below. Refer to Update the solution for details on the migration outcome.

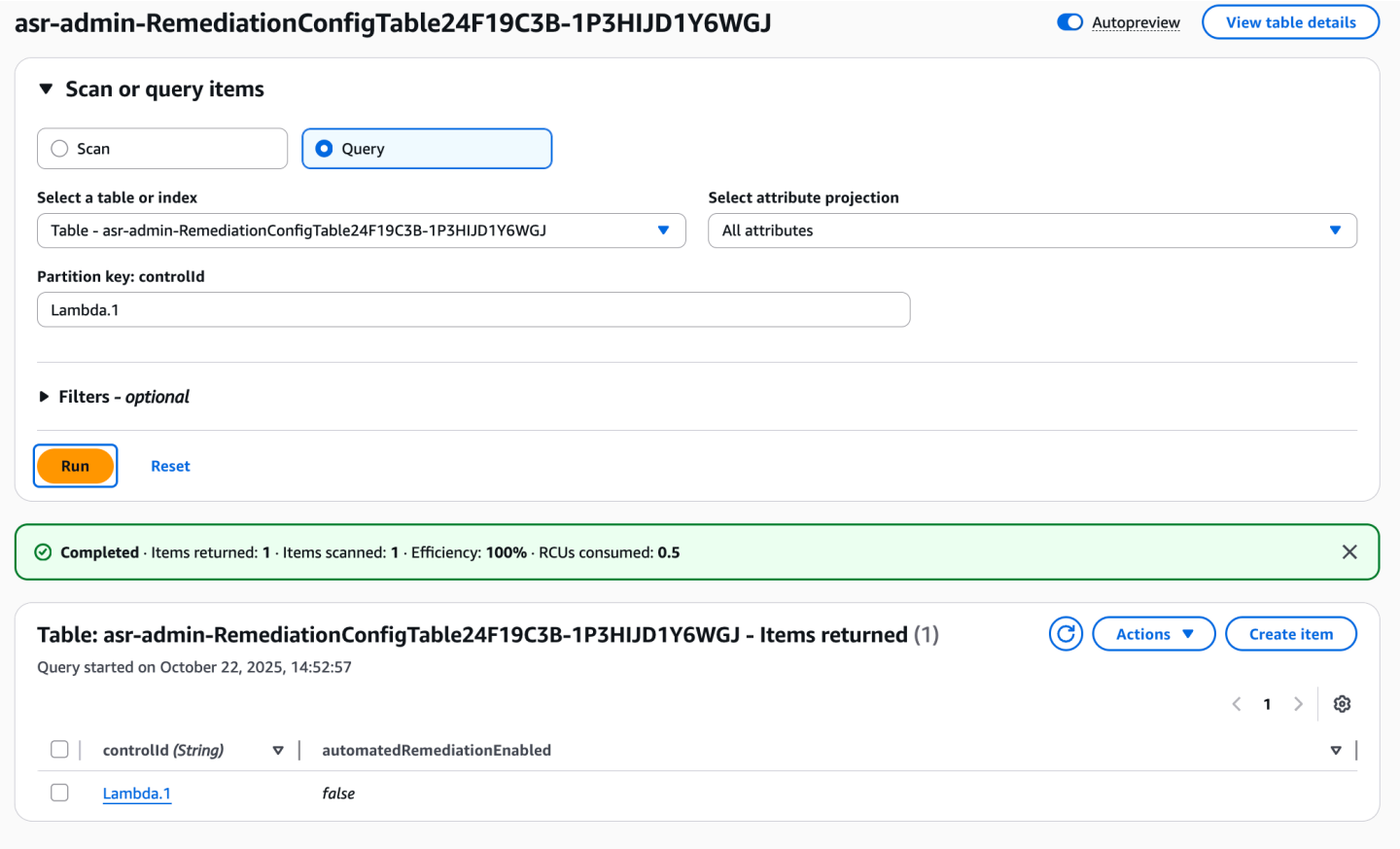
Account	Purpose	Action in us-east-1	Action in us-west-2
111111111111	Admin	Locate the Remediation Configuration DynamoDB table.	None
222222222222	Member	None	None

Modify the Remediation Configuration Table

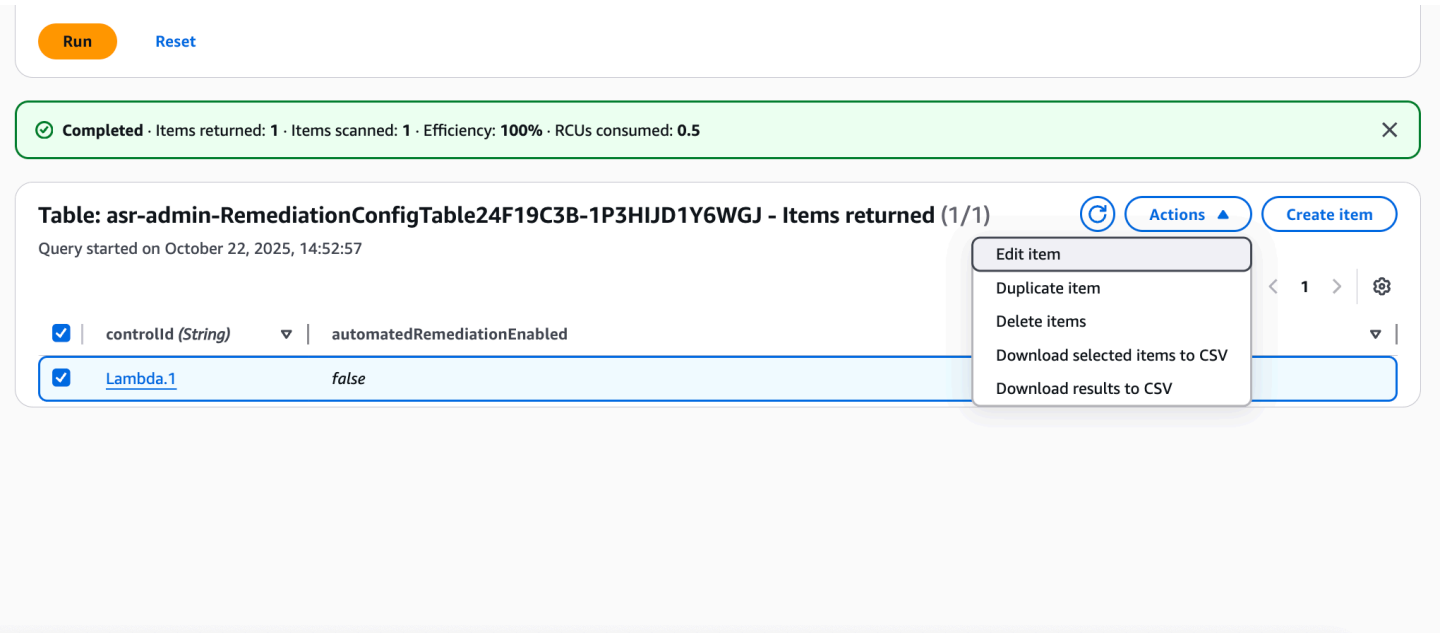
In the DynamoDB console where you have located the Remediation Configuration table, choose **Explore Table Items**.

Each item in the table corresponds to a Security Hub control supported by the solution. Each item has a `automatedRemediationEnabled` attribute that can be modified to enable fully automated remediations for the associated control.

To enable Lambda.1, under **Scan or query items** choose **Query**. Under **Partition key: controlId** enter `Lambda.1` and choose **Run**. You will see a single item returned corresponding to the Lambda.1 control.



Now, select the `Lambda.1` item then choose **Actions > Edit item**.



Finally, change the `automatedRemediationEnabled` attribute value to **True**. Choose **Save and Close**.

Account	Purpose	Action in us-east-1	Action in us-west-2
111111111111	Admin	Modify the Remediation Configuration DynamoDB table.	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 might 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	None	None

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

(Optional) Filter fully automated remediations

To limit the scope in which the solution runs fully automated remediations, apply **resource filters**. Filters apply only to fully automated remediations and do not affect manually initiated remediations.

A resource filter is a reusable definition that scopes which findings are automatically remediated. Each filter can combine account IDs, AWS Organizations organizational units (OUs), resource ARN patterns, and resource tags. You create filters on the **Filters** page of the Web UI, then apply them to a single control from the **Controls** page (or to every control at once) using either **Include** or **Exclude** mode.

For the full walkthrough, including creating filters, applying them to controls, and how the solution evaluates include and exclude modes, see Create and apply resource filters in the *Administrator guide*.

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 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 Confirm member roles stack deleted	None

Account	Purpose	Action in us-east-1	Action in us-west-2
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 (for example, 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 AWS KMS keys for deletion

The admin and member stacks both create and retain an AWS KMS key. You 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	Identify and schedule member key for deletion

Account	Purpose	Action in us-east-1	Action in us-west-2
		Identify and schedule member key for deletion	
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) v1.0.0, 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 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 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 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 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 Ensure CloudTrail log file validation is activated	CloudTrail.4	2.2	CloudTrail.3	3.2	CloudTrail.4		CloudTrail.4
ASR-EnableCloudTrailToCloudWatchLogging 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 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 CodeBuild project environment variables should not contain clear text credentials	CodeBuild.2		CodeBuild.2		CodeBuild.2		CodeBuild.2
ASR-EnableAWSConfig 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 Amazon EBS snapshots should not be publicly restorable	EC2.1		EC2.1		EC2.1		EC2.1
ASR-RemoveVPCDefaultSecurityGroupRules 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 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 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 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 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 User credentials should be turned off if not used within 90 days	IAM.8	1.3	IAM.7		IAM.8		IAM.8
ASR-RevokeUnusedIAMUserCredentials 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 Lambda functions should prohibit public access	Lambda.1		Lambda.1		Lambda.1		Lambda.1
ASR-MakeRDSSnapshotPrivate 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 RDS DB Instances should prohibit public access	RDS.2		RDS.2		RDS.2	2.3.3	RDS.2
ASR-EncryptRDSSnapshot 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 RDS DB instances should be configured with multiple Availability Zones	RDS.5				RDS.5		RDS.5
ASR-EnableEnhancedMonitoringOnRDSInstance 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 RDS clusters should have deletion protection activated	RDS.7				RDS.7		RDS.7
ASR-EnableRDSInstanceDeletionProtection 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 RDS automatic minor version upgrades should be activated	RDS.13				RDS.13	2.3.2	RDS.13
ASR-EnableCopyTagsToSnapshotOnRDSCluster 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 Amazon Redshift clusters should prohibit public access	Redshift.1		Redshift.1		Redshift.1		Redshift.1
ASR-EnableAutomaticSnapshotsOnRedshiftCluster 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 Amazon Redshift clusters should have audit logging activated	Redshift.4				Redshift.4		Redshift.4
ASR-EnableAutomaticVersionUpgradeOnRedshiftCluster 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 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 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 S3 buckets should prohibit public write access		S3.3					S3.3
ASR-EnableDefaultEncryptionS3 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 S3 buckets should require requests to use SSL	S3.5		S3.5	2.1.2	S3.5	2.1.1	S3.5
ASR-S3BlockDenylist 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 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 Ensure S3 bucket access logging is activated on the CloudTrail S3 bucket		2.6					CloudTrail.7
ASR-EnableKeyRotation 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 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 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 Ensure a log metric filter and alarm exist for usage of the "root" user		3.3	CW.1	4.3			CloudWatch.3
ASR-CreateLogMetricFilterAndAlarm 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 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 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 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 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 Ensure a log metric filter and alarm exist for AWS Config configuration changes		3.9		4.9			CloudWatch.9
ASR-CreateLogMetricFilterAndAlarm 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 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 Ensure a log metric filter and alarm exist for changes to network gateways		3.12		4.12			CloudWatch.12
ASR-CreateLogMetricFilterAndAlarm 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 Ensure a log metric filter and alarm exist for VPC changes		3.14		4.14			CloudWatch.14
AWS-DisablePublicAccessForSecurityGroup 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 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 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 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 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 CloudFront distributions should not point to non-existent S3 origins	CloudFront.12				CloudFront.12		CloudFront.12
ASR-RemoveCodeBuildPrivilegedMode 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 Stopped EC2 instances should be removed after a specified time period	EC2.4				EC2.4		EC2.4
ASR-EnableIMDSV2OInstance 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 Security groups should only allow unrestricted incoming traffic for authorized ports	EC2.18				EC2.18		EC2.18

Description	AWS FSBP	CIS v1.2.0	PCI v3.2.1	CIS v1.4.0	NIST	CIS v3.0.0	Security control ID
ASR-DisableUnrestrictedAccessToHighRiskPorts 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 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 ECR private repositories should have image scanning configured	ECR.1				ECR.1		ECR.1
ASR-EnableGuardDuty 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 S3 bucket server access logging should be enabled	S3.9				S3.9		S3.9
ASR-EnableBucketEventNotifications 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 S3 buckets should have lifecycle policies configured	S3.13				S3.13		S3.13
ASR-EnableAutoSecretRotation 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 Remove unused Secrets Manager secrets	SecretsManager.3				SecretsManager.3		SecretsManager.3
ASR-UpdateSecretRotationPeriod 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 API Gateway REST API cache data should be encrypted at rest					APIGateway.5		APIGateway.5
ASR-SetLogGroupRetentionDays 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 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 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 GuardDuty detectors should be tagged							GuardDuty.4
ASR-AttachSSMPermissionsToEC2 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 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 Amazon Macie should be enabled	Macie.1				Macie.1		Macie.1
ASR-EnableAthenaWorkGroupLogging 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 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 ECS containers should be limited to read-only access to root filesystems	ECS.5				ECS.5		ECS.5
ASR-EnableElastiCacheBackups 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 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 ElastiCache replication groups should have automatic failover enabled	ElastiCache.3				ElastiCache.3		ElastiCache.3
ASR-ConfigureDynamoDBAutoScaling 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 DynamoDB tables should be tagged							DynamoDB.5
ASR-EnableDynamoDDeletionProtection 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.

AI Toolkit for Custom Remediations

To accelerate the workflows described in this section, the solution provides the AI Toolkit for Custom Remediations. The toolkit packages ASR's remediation best practices, guardrails, and development patterns as an instruction prompt that you provide to an AI assistant in your integrated development environment (IDE). With this context, the assistant can help you author ASR-compatible remediation runbooks, control runbooks, IAM roles, and CDK constructs that follow the naming conventions and integration steps documented in the following sections — producing production-ready custom remediations with fewer iterations and greater confidence.

The toolkit is optional and does not change how remediations are built or deployed; it is a development aid that makes the solution's conventions and constraints visible to your AI assistant. You remain responsible for reviewing and testing any generated code before deploying it.

Overview of the manual 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 Amazon EventBridge. 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 (such as 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-runbook-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. Update the 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
/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

Name	Value	Use
/ASR/Filters/AccountFilterMode	Include, Exclude, or Disabled	Controls the Account ID filtering behavior for fully automated remediations
/ASR/Filters/AccountFilters	Comma-delimited list of AWS Account IDs	List of AWS Account IDs for which the solution should filter automated remediations.
/ASR/Filters/OUFilterMode	Include, Exclude, or Disabled	Controls the Organizational Units (OUs) filtering behavior for fully automated remediations
/ASR/Filters/OUFilters	Comma-delimited list of Organization Unit Ids	List of OUs for which the solution should filter automated remediations.
/ASR/Filters/TagFilterMode	Include, Exclude, or Disabled	Controls the Resource Tag filtering behavior for fully automated remediations
/ASR/Filters/TagFilters	Comma-delimited list of Resource Tag Keys	List of Resource Tag Keys for which the solution should filter automated remediations.

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, choose **Edit**.
3. Expand "Subscription filter policy" and toggle the "Subscription filter policy" option to enable filters.
4. Choose 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, complete the following steps:

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. It specifies the runbook that runs when you remediate this AWS Config rule. RunbookRole is the role that the orchestrator assumes when running this remediation. It is not a required field, and if left out, the orchestrator defaults to using the account's member role.
2. Once this is in place, you can remediate your AWS Config rule using the "Remediate with ASR" custom action found on the Security Hub.

Web UI

The solution's Web UI allows users to remediate AWS Security Hub findings in one-click, view and download past remediations, and delegate access to the solution.

The Web UI is not required to use the solution; you can alternatively configure fully-automated remediations to avoid the need for manual execution, or leverage the AWS Security Hub CSPM console to kick-off remediations using the **Remediate with ASR** custom action.

> **Note**
>
> You must set the `ShouldDeployWebUI` parameter to "yes" when deploying the Admin stack in order to use the solution's Web UI.

How it works

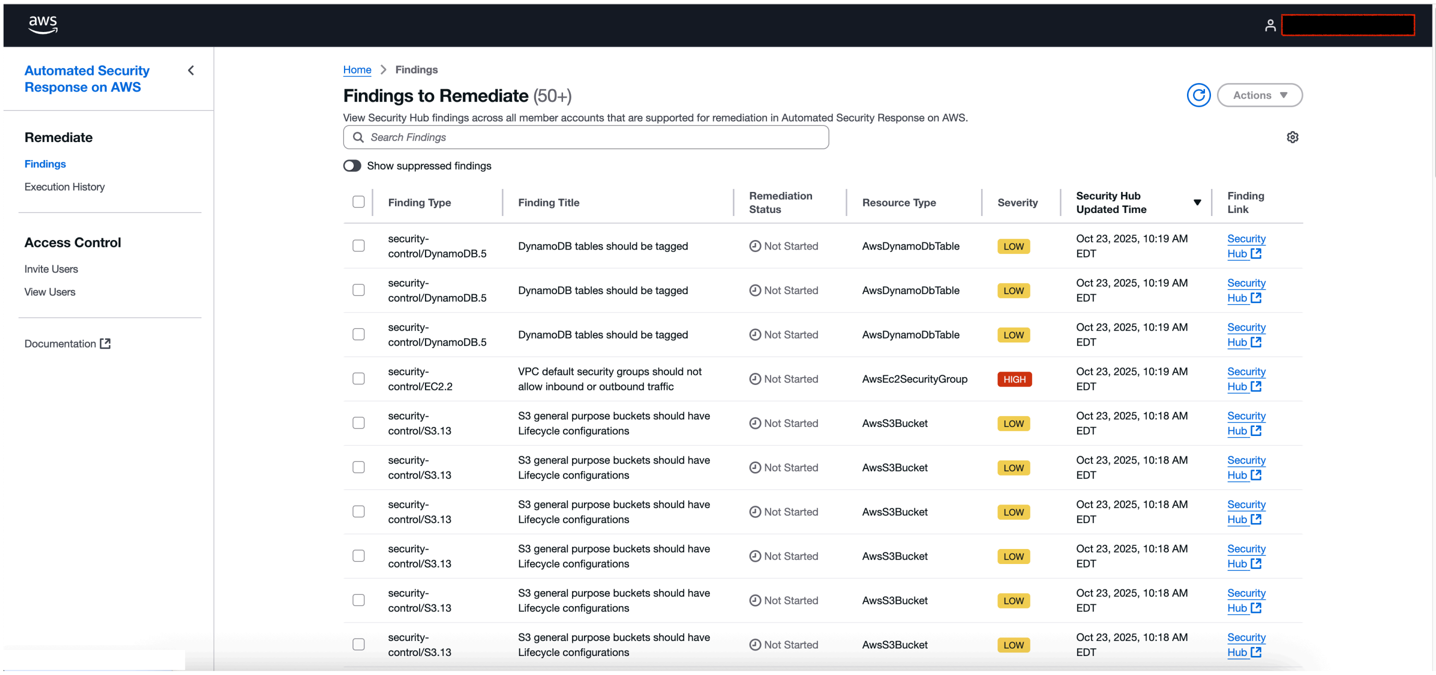
The solution's Web User Interface is a Single-Page Web Application hosted in your account by Amazon S3 and distributed by Amazon CloudFront. The solution also deploys a REST API using API Gateway to support operations in the Web UI.

When the Admin stack is deployed, the solution's Lambda functions begin loading all AWS Security Hub findings supported by the solution that are present in your Admin account into DynamoDB. Once this is complete, the Findings presented in the Web UI are kept in sync with Security Hub in near real-time thanks to the EventBridge rules deployed by the solution.

Every week, the solution triggers its Lambda functions to refresh the DynamoDB table storing AWS Security Hub findings displayed in the Web UI. This is designed to clean up stale data and keep the DynamoDB tables up to date. If you want to configure this

baseline to run more or less often, modify the EventBridge Rule named `SO0111-ASR-SynchronizationFindingsLambdaWeeklyRule` located in your Admin account in the same region where you deployed the solution.

Run remediations directly in the Web UI



On the **Findings** page, Admin or Delegated Admin users can view all AWS Security Hub findings supported by the solution for remediation. This includes findings for Security Hub member accounts onboarded with the Security Hub primary account. If the solution is also deployed in the aggregation region, then findings in any onboarded region will also be displayed. To view the list of findings supported by the solution, see the playbooks section.

Account Operator users will only be able to view findings which originate in AWS Accounts that they have access to as defined in their invitation. Additionally, they will only be able to run remediations for resources in the accounts they are associated with.

To run remediations, select any number of items in the table and choose **Actions > Remediate**. You can also **suppress** findings by choosing **Actions > Suppress**, which hides the selected findings from the default view. You can view suppressed findings at any time by choosing the **Show suppressed findings** toggle.

Once you have begun remediation for a finding, you can choose the **Remediation Status** column while the remediation is either `In Progress` or `Failed` to be taken directly to that remediation on the **Execution History** page.

Filter available findings and remediations

On both the **Findings** and **Execution History** pages, you can filter the data displayed in the table by any of the columns present in each respective table.

For example, on the **Findings** page, you might filter on **Finding Type** to search for specific kinds of AWS Security Hub findings (such as Lambda.1 or Athena.4) by choosing the search bar and selecting **Finding Type**.

> **Note**
>
> Values that are autopopulated in the search bar do not represent a comprehensive list of available data. The suggested values for each search criteria only represent the data currently fetched and displayed in the UI.

You may also combine multiple attributes in a single search. For example, you can apply both **Finding Type** and **Resource ID** in your search to perform a logical AND query. Additionally, you can apply multiple of the same filter criteria to perform a logical OR search, such as **Finding Type = Lambda.1** and **Finding Type = Athena.4**. The same principles apply to the **Execution History** page

Authentication & Authorization in the Web UI

The solution's Web UI is protected by authentication provided by Amazon Cognito. When the solution is deployed, a Cognito User Pool, Cognito App Client, and Cognito User Pool Domain are provisioned and configured alongside the Web UI. The email address provided as a parameter to the Admin stack is assigned temporary credentials and is given Administrator access to the Web UI.

There are three permission types that define a user's access to the Web UI:

Permission Type	Access Level	Use Case
Admin	Full control in the Web UI; Can view all findings and remediations, run any remediation, and invite/view any user.	Assigned only to the user who deployed the Admin stack when they provide their email address during CloudFormation deployment.

Permission Type	Access Level	Use Case
Delegated Admin	Elevated control in the Web UI; Can view all findings and remediations, run any remediation, and invite/view Account Operator users. Cannot invite or view Admins and Delegated Admins in the Web UI.	The Admin user can delegate access to the solution by inviting Delegated Admin users, who will be able to run and manage any remediations.
Account Operator	Limited control in the Web UI; Restricted to view and remediate findings only in accounts they are associated with upon invitation. Cannot invite or view additional users.	Day-to-day users who should have limited access to run remediations in a subset of onboarded accounts. Admins or Delegated Admins are responsible for inviting these users and defining their scope.

All users must be invited by an Admin or Delegated Admin before they are able to sign in to the Web UI. To invite additional users, an Admin or Delegated Admin can enter their email address and permission level on the **Invite Users** page of the Web UI.

Admins and Delegated Admins can also view, manage, and delete existing users. To see a list of all users, navigate to the **View Users** page.

To manage an existing user, select the user from the table and choose **Manage User**. You can then delete the user by choosing **Delete User**. If the user is an Account Operator, you can modify the list of AWS Account IDs they have access to in the context of the solution. Changing the permission type for an existing user is not currently supported.

Delegated Admins can only view and manage Account Operator users.

Integrating with external IdPs

You can customize the authentication mechanism provided by the solution to allow users to sign-in using your own OIDC or SAML identity provider, such as Okta or Microsoft Entra ID. The following

steps for integrating with external IdPs requires access to the AWS Account where the Admin stack is deployed.

> **Important**
>
> Users must still be invited prior to signing in using any external IdP you configure to work with solution. Additionally, the email address linked to their IdP profile must match the email provided in their invitation.

Step 1 - Locate the solution's user pool

In the Amazon Cognito console, locate the solution's user pool named **SO0111-ASR-UserPool**.

Choose the user pool name **SO0111-ASR-UserPool** to be taken to the **overview** page. From there, choose **Social and external providers** from the navigation bar.

Step 2 - Add your identity provider

On the **Social and external providers** page, choose the **Add identity provider** button on the top right.

Select either **OIDC** or **SAML**, depending on your identity provider.

Once you select your provider type, you will be prompted to enter information about your identity provider.

Fill in the following fields for **SAML** providers:

1. **Provider name**: A friendly name for your provider
2. **IdP-initiated SAML sign-in**: Select `Require SP-initiated SAML assertions - Recommended`
3. **Metadata document source**: Select `Upload metadata document`
4. **Metadata document**: Upload your SAML metadata document provided by your IdP.
5. Under **Map attributes between your SAML provider and your user pool** choose **Add another attribute**. For **User pool attribute** select `email` from the dropdown. For **SAML attribute**, enter the full name of the attribute where the user's email address is stored in your SAML identity provider. For example, `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress`.

6. Choose **Add identity provider** to save your changes.

Fill in the following fields for **OIDC** providers:

1. **Provider name**: A friendly name for your provider
2. **Client ID**: Enter the client ID provided by your OpenID Connect identity provider.
3. **Client secret**: Enter the client secret provided by OpenID Connect identity provider.
4. **Authorized scopes**: Enter `openid profile email`
5. **Attribute request method**: Select `GET` or `POST` based on your identity provider's configuration.
6. **Setup method**: Select `Auto fill through issuer URL` and enter the **Issuer URL** from your OIDC provider. Alternatively, enter the values manually.
7. Under **Map attributes between your OpenID Connect provider and your user pool** choose **Add another attribute**. For **User pool attribute** select `email` from the dropdown. For **OpenID Connect attribute**, enter the full name of the attribute where the user's email address is stored in your OIDC identity provider. For example, `email`.
8. Choose **Add identity provider** to save your changes.

> **Important**
>
> You must add an attribute mapping for the `email` user pool attribute, even if your identity provider's attribute name is also `email`.

Step 3 - Add your provider to the solution's App Client

Navigate to the **App Clients** page and select the client named **SO0111-ASR-WebUI-UserPoolClient**.

Choose the **Login Pages** tab and under **Managed login pages configuration** choose **Edit**.

In the **Identity providers** field, add the identity provider you created in the previous step. Choose **Save Changes**.

Step 4 - Configure your identity provider

To allow your identity provider to redirect to the solution's Web UI after login, you must allowlist the following URLs in your IdP configuration.

Depending on your provider type, allowlist one of the following Callback URLs:

1. SAML Callback URL: https://so0111-asr-*<your-aws-account-id>*.auth.*<aws-region>*.amazoncognito.com/saml2/idpresponse
2. OIDC Callback URL: https://so0111-asr-*<your-aws-account-id>*.auth.*<aws-region>*.amazoncognito.com/oauth2/idpresponse

You should replace `<your-aws-account-id>` with the AWS Account ID where you have deployed the Admin stack, and `<aws-region>` with the region where you deployed the Admin stack.

Step 5 - Verify your integration

Navigate to the Web UI login page. Confirm that your custom identity provider is visible on the login page.

To test the integration, invite a new user using the **Invite Users** page. Then, ensure the user can authenticate by choosing your custom identity provider on the Web UI login page.

The user's profile in your custom IdP must be linked to the same email address provided in their invitation. In other words, the email address in your provider's claims must match the invitation.

Reference

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

Data collection

This solution sends operational metrics to AWS (the "Data") about the use of this solution. We use this Data to better understand how customers use this solution and related services and products. AWS's collection of this Data is subject to the AWS Privacy Notice.

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
- Manish Jangid
- Andrew Stephen
- Peter DeVries
- Mukta Dadariya
- Immanuel George
- Harika Kondakindi
- Narmeen Ali

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 Apache License Version 2.0 available at The Apache Software Foundation.