aws

User Guide

AWS Resilience Hub

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Resilience Hub: User Guide

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

AWS Resilience Hub 1
What is AWS Resilience Hub? 1
AWS Resilience Hub – Resilience management 2
AWS Resilience Hub – Resilience testing 6
AWS Resilience Hub concepts 7
AWS Resilience Hub personas 11
Supported AWS Resilience Hub resources 12
AWS Resilience Hub and myApplications 16
Getting started 18
Prerequisites 18
Add an application 19
Using AWS Resilience Hub 32
AWS Resilience Hub summary 32
AWS Resilience Hub dashboard 37
Managing applications 39
Managing resiliency policies 65
Managing resiliency assessments in AWS Resilience Hub 71
Managing resiliency assessments from Resiliency widget 82
Managing alarms 85
Managing standard operating procedures 92
Managing AWS Fault Injection Service experiments 97
Understanding resiliency scores 107
Integrating recommendations into applications 122
Using AWS Resilience Hub APIs to describe and manage application 127
Preparing the application 128
Running and analyzing the application 134
Modify your application 152
Security 156
Data protection 157
Identity and Access Management 158
Infrastructure security 211
Resilience Checks for AWS services 211
Amazon Elastic File System 212
Amazon Relational Database Service and Amazon Aurora 213

Amazon Simple Storage Service ... 214
Amazon DynamoDB ... 215
Amazon Elastic Compute Cloud ... 216
Amazon EBS ... 217
AWS Lambda ... 217
Amazon Elastic Kubernetes Service ... 218
Amazon Simple Notification Service ... 219
Amazon Simple Queue Service ... 219
Amazon Elastic Container Service ... 219
Elastic Load Balancing ... 220
Amazon API Gateway ... 220
Amazon DocumentDB ... 221
NAT Gateway ... 221
Amazon Route 53 ... 221
Amazon Application Recovery Controller (ARC) ... 222
Amazon FSx for Windows File Server ... 222
AWS Step Functions ... 223
Amazon ElastiCache (Redis OSS) ... 223
Working with other services ... 224
AWS CloudFormation ... 225
AWS CloudTrail ... 226
AWS Systems Manager ... 226
AWS Trusted Advisor ... 226
Document history ... 229
Next generation Resilience Hub ... 266
What is Next generation Resilience Hub? ... 266
Key capabilities at a glance ... 267
Supported Regions and availability ... 267
Pricing overview ... 268
Related services ... 269
Setting up Next generation Resilience Hub ... 270
Prerequisites ... 270
Required IAM permissions and roles ... 271
AWS managed policies ... 282
Configuring AWS Organizations integration (optional) ... 293
Multi-account setup (without AWS Organizations) ... 293

Customer-managed keys (optional) ... 295
Getting started with Next generation Resilience Hub ... 296
Tutorial overview and prerequisites ... 296
Step 1: Configure a resilience policy ... 297
Step 2: Create your first system and service ... 298
Step 3: Run your first failure mode assessment ... 300
Step 4: Review failure mode findings and recommendations ... 301
Step 5: Enable dependency discovery (optional) ... 302
Step 6: Run your first resilience test (optional) ... 302
Systems, user journeys, and services ... 303
Overview: the systems, user journeys, and services hierarchy ... 304
Creating and managing systems ... 305
Creating and managing user journeys ... 307
Creating and managing services ... 310
Adding and removing resources from a service ... 315
Importing application context from external registries ... 316
Viewing service topology ... 317
Example: Modeling a multi-account application ... 317
Resilience policies ... 320
Understanding resilience policies ... 320
Policy components ... 321
Creating a resilience policy ... 321
Applying policies at user journey and service levels ... 322
Policy inheritance and multi-level evaluation ... 323
Managing and updating policies ... 324
Sharing a policy across your organization ... 325
Assessment ... 326
Dependency discovery ... 326
Failure mode assessments ... 338
Resilience testing ... 352
How resilience testing works ... 352
Configuring a test ... 353
Available tests ... 355
Test runs and report ... 367
IAM execution roles for resilience testing ... 368
Dashboard and reporting ... 408

Central SRE dashboard overview ... 408
Service-level view ... 409
Filtering by policy, account, Region, and system ... 409
Generating compliance and resilience posture reports ... 410
Working with AWS DevOps Agent in the next generation of Resilience Hub ... 411
Prerequisites ... 412
Add DevOps Agent permissions to the invoker role ... 413
Connect a service to the Agent Space ... 414
Investigate findings and test runs ... 415
AWS Organizations integration ... 416
Overview of multi-account governance in the next generation of Resilience Hub ... 416
Setting up Organizations integration ... 417
Service-Linked Roles ... 419
Applying resilience policies across accounts ... 419
Tracking changes to a shared policy ... 419
Viewing organization-wide resilience posture ... 420
Managing member account onboarding ... 421
Required IAM permissions for delegated administrator setup ... 421
Migrating from AWS Resilience Hub ... 422
What changes with Next generation Resilience Hub ... 423
Concept mapping: AWS Resilience Hub v1 to Next generation Resilience Hub ... 424
Step-by-step migration guide ... 425
Pricing transition ... 426
Known limitations during migration ... 427
Security ... 427
IAM roles and permissions reference ... 428
Data encryption ... 434
VPC endpoints ... 450
CloudTrail integration ... 450
Compliance considerations ... 450
Monitoring ... 451
CloudWatch metrics reference ... 452
CloudTrail event logging ... 453
EventBridge notifications ... 455
Setting up CloudWatch alarms for Next generation Resilience Hub ... 462
Quotas and limits ... 463

Service quotas ... 463
Requesting quota increases ... 465
Troubleshooting ... 465
Dependency discovery issues ... 466
Failure mode assessment issues ... 467
IAM and permissions errors ... 469
AWS Organizations configuration issues ... 470
Resource discovery issues ... 471
Resilience testing issues ... 475
Known issues and workarounds ... 477
API reference ... 477
Making API requests and authentication ... 478
Actions ... 478
Data types ... 485
Error codes ... 488
Document history ... 489
AWS Glossary ... 490

AWS Resilience Hub

AWS Resilience Hub is a central location for you to manage and improve the resilience posture of your applications on AWS. AWS Resilience Hub enables you to define your resilience goals, assess your resilience posture against those goals, and implement recommendations for improvement.

Topics

- What is AWS Resilience Hub?
- Getting started
- Using AWS Resilience Hub
- Using AWS Resilience Hub APIs to describe and manage application
- Security in AWS Resilience Hub
- Resilience Checks for AWS services
- Working with other services
- Document history for the AWS Resilience Hub User Guide

What is AWS Resilience Hub?

AWS Resilience Hub is a central location for you to manage and improve the resilience posture of your applications on AWS. AWS Resilience Hub enables you to define your resilience goals, assess your resilience posture against those goals, and implement recommendations for improvement based on the AWS Well-Architected Framework. Within AWS Resilience Hub, you can also create and run AWS Fault Injection Service experiments, which mimic real-life disruptions to your application to help you better understand dependencies and uncover potential weaknesses. AWS Resilience Hub provides a central place with all the AWS services and tools that you need to continuously strengthen your resilience posture. AWS Resilience Hub works with other services to provide recommendations and help you to manage your application resources. For more information, see Working with other services.

The following table provides the documentation links of all the related resiliency services.

Related AWS resiliency services and references

AWS resiliency service	Documentation link
AWS Elastic Disaster Recovery	What is Elastic Disaster Recovery
AWS Backup	What is AWS Backup
Amazon Application Recovery Controller (ARC) (ARC)	What is Amazon Application Recovery Controller (ARC)

Topics

- AWS Resilience Hub – Resilience management
- AWS Resilience Hub – Resilience testing
- AWS Resilience Hub concepts
- AWS Resilience Hub personas
- AWS Resilience Hub supported resources
- AWS Resilience Hub and myApplications

AWS Resilience Hub – Resilience management

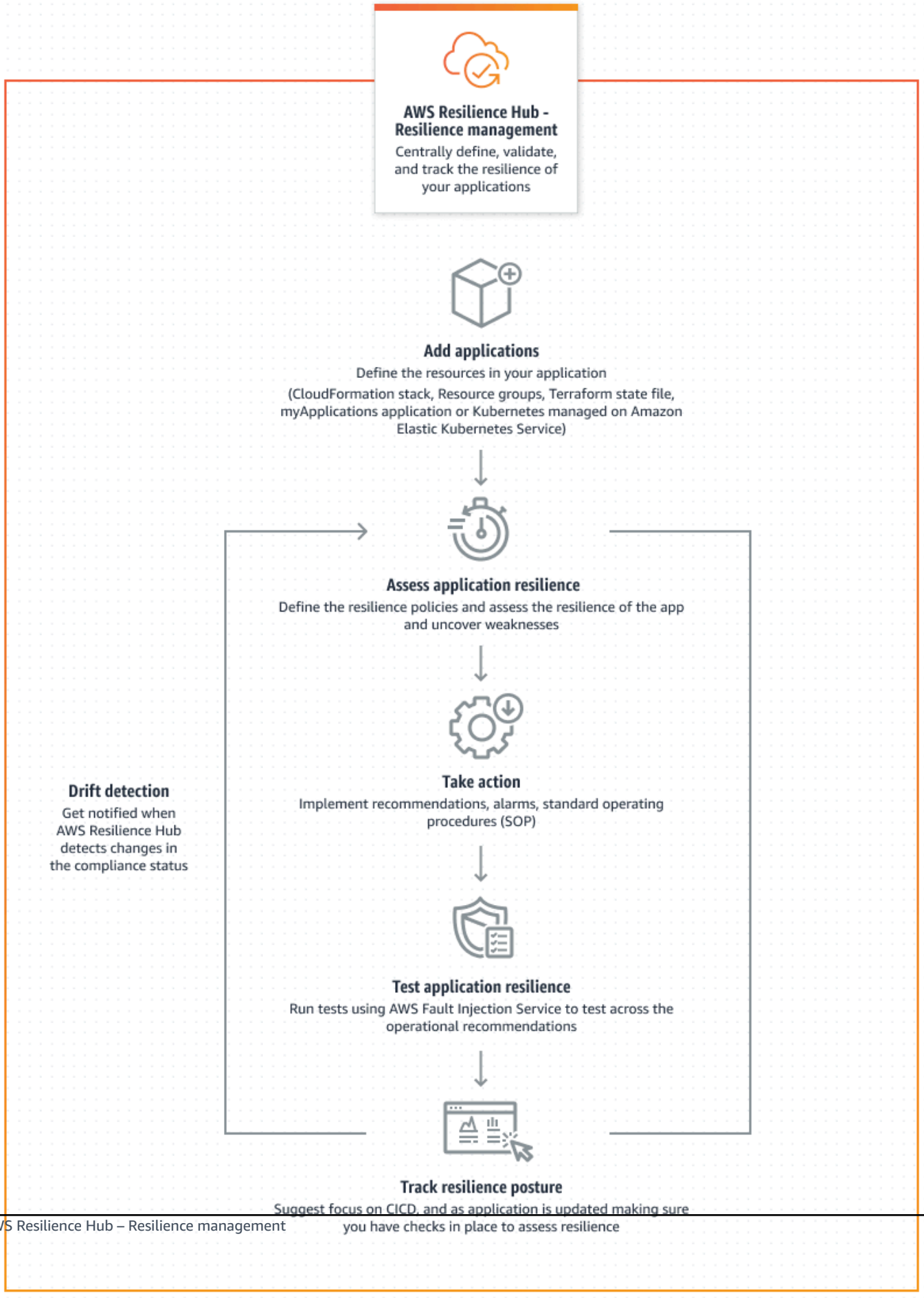
AWS Resilience Hub gives you a central place to define, validate, and track the resiliency of your AWS application. AWS Resilience Hub helps you to protect your applications from disruptions, and reduce recovery costs to optimize business continuity to help meet compliance and regulatory requirements. You can use AWS Resilience Hub to do the following:

- Analyze your infrastructure and get recommendations to improve the resiliency of your applications. In addition to architectural guidance for improving your application resiliency, the recommendations provide code for meeting your resiliency policy, implementing tests, alarms, and standard operating procedures (SOPs) that you can deploy and run with your application in your integration and delivery (CI/CD) pipeline.
- Evaluate recovery time objective (RTO) and recovery point objective (RPO) targets under different conditions.
- Optimize business continuity while reducing recovery costs.
- Identify and resolve issues before they occur in production.

After you deploy an application into production, you can add AWS Resilience Hub to your CI/CD pipeline to validate every build before it is released into production.

How AWS Resilience Hub works

The following diagram provides a high-level outline of how AWS Resilience Hub works.


AWS Resilience Hub - Resilience management
Centrally define, validate, and track the resilience of your applications
Add applications
Define the resources in your application
(CloudFormation stack, Resource groups, Terraform state file, myApplications application or Kubernetes managed on Amazon Elastic Kubernetes Service)
Assess application resilience
Define the resilience policies and assess the resilience of the app and uncover weaknesses
Take action
Implement recommendations, alarms, standard operating procedures (SOP)
Test application resilience
Run tests using AWS Fault Injection Service to test across the operational recommendations
Track resilience posture
Suggest focus on CICD, and as application is updated making sure you have checks in place to assess resilience
Drift detection
Get notified when AWS Resilience Hub detects changes in the compliance status

Describe

Describe your application by importing resources from AWS CloudFormation stacks, Terraform state files, AWS Resource Groups, Amazon Elastic Kubernetes Service clusters, or you can choose from applications that are already defined in myApplications.

Define

Define the resilience policies for your applications. These policies include RTO and RPO targets for applications, infrastructure, Availability Zone, and Region disruptions. These targets are used to estimate whether the application meets the resiliency policy.

Assess

After you describe your application and attach a resiliency policy to it, run a resiliency assessment. The AWS Resilience Hub assessment uses best practices from the AWS Well-Architected Framework to analyze the components of an application and uncover potential resilience weaknesses. These weaknesses can be caused by incomplete infrastructure setup, misconfiguration, or situations where additional configuration improvements are needed. To improve resiliency, update your application and resiliency policy according to the recommendations from the assessment report. Recommendations include configurations of components, alarms, tests, and recovery SOPs. Then, you can run another assessment and compare the results with the previous report to see how much resiliency improves. Reiterate this process until your estimated workload RTO and estimated workload RPO meets your RTO and RPO targets.

Validate

Run tests to measure the resiliency of your AWS resources and the amount of time it takes to recover from application, infrastructure, Availability Zone, and AWS Region incidents. To measure resiliency, these tests simulate outages of your AWS resources. Examples of outages include network unavailable errors, failovers, stopped processes, Amazon RDS boot recovery, and problems with your Availability Zone.

View and track

After you deploy an AWS application into production, you can use AWS Resilience Hub to continue tracking the resiliency posture of the application. If an outage occurs, the operator can view the outage in AWS Resilience Hub and launch the associated recovery process.

AWS Resilience Hub – Resilience testing

AWS Resilience Hub supports an enhanced integration with the AWS FIS. This integration allows AWS Resilience Hub to offer tailored recommendations using AWS FIS actions and scenarios based on the specific context of the application being assessed. Running the recommended experiments or conducting your own tests using the AWS FIS service will directly contribute to improving your application's resilience score.

These AWS FIS actions and scenarios test an application's resiliency posture by creating disruptive events so that you can observe how your application responds. AWS FIS provides multiple pre-built scenarios and large selection of actions that generate disruptions. In addition, it also includes controls and guardrails that you need to run the experiments in production. The controls and guardrails include options to perform automatic roll back or stop the experiment if specific conditions are met. To get started using the AWS FIS to run experiments from AWS Resilience Hub console, complete the prerequisites that are defined in the section called “Prerequisites” section.

The following table lists all the available AWS FIS options from the navigation pane and the links to the associated AWS FIS documentation that contains the procedures to start using AWS FIS tests from AWS Resilience Hub console.

AWS FIS navigation menu options and references

AWS FIS navigation menu option	AWS FIS documentation
Resilience testing	Create an experiment template
Scenario library	AWS FIS library
Experiment templates	Experiment templates for AWS FIS

The following table lists all the available AWS FIS options from the dropdown menu in **Resilience testing** section and the links to the associated AWS FIS documentation that contains the procedures to start using AWS FIS tests from AWS Resilience Hub console.

AWS FIS dropdown menu options and references

AWS FIS dropdown menu option	AWS FIS documentation
Create experiment template	Create an experiment template

AWS FIS dropdown menu option	AWS FIS documentation
Create an experiment from scenario	Using a scenario

AWS Resilience Hub concepts

These concepts can help you better understand the AWS Resilience Hub's approach to helping improve application resiliency and prevent application outages.

Resiliency

The ability to maintain availability and to recover from software and operational disruption in a designated time frame.

Recovery point objective (RPO)

The maximum acceptable amount of time since the last data recovery point. This determines what is considered an acceptable loss of data between the last recovery point and the interruption of service.

Recovery time objective (RTO)

The maximum acceptable delay between the interruption of service and restoration of service. This determines what is considered an acceptable time window when service is unavailable.

Estimated workload recovery time objective

The estimated workload recovery time objective (estimated workload RTO) is the RTO that your application is estimated to meet based on the imported application definition and then run an assessment.

Estimated workload recovery point objective

The estimated workload recovery point objective (estimated workload RPO) is the RPO that your application is estimated to meet based on the imported application definition and then run an assessment.

Application

An AWS Resilience Hub application is a collection of AWS supported resources that are continuously monitored and assessed to manage its resiliency posture.

Application Component

A group of related AWS resources that work and fail as a single unit. For example, if you have a primary and replica database, then both databases belong to the same Application Component (AppComponent).

AWS Resilience Hub determines which AWS resources can belong to which type of AppComponent. For example, a `DBInstance` can belong to `AWS::ResilienceHub::DatabaseAppComponent` but not to `AWS::ResilienceHub::ComputeAppComponent`.

Application compliance status

AWS Resilience Hub reports the following compliance status types for your applications.

Policy met

The application is estimated to meet its RTO and RPO targets defined in the policy. All its components meet the defined policy objectives. For example, you selected an RTO and RPO target of 24 hours for disruptions across AWS Regions. AWS Resilience Hub can see that your backups are copied to your fallback Region. You are still expected to maintain a recover from a backup standard operating procedure (SOP), and to test and time it. This is in the operational recommendations and part of your overall resiliency score.

Policy breached

The application could not be estimated to meet the RTO and RPO targets defined in the policy. One or more of its AppComponents do not satisfy the policy objectives. For example, you selected an RTO and RPO target of 24 hours for disruptions across AWS Regions, but your database configuration does not include any cross-Region recovery method, such as a global replication and backup copies.

Not assessed

The application requires an assessment. It's not currently assessed or tracked.

Changes detected

There is a new published version of the application that has not yet been assessed.

Drift detection

AWS Resilience Hub runs drift notification while running an assessment for your application to check if the changes in AppComponent configurations have affected the compliance status of your application. In addition, it also checks and detects changes such as addition or deletion of resources within the application's input sources and notifies about the same. For comparison, AWS Resilience Hub uses the previous assessment in which the application component met the policy. AWS Resilience Hub detects the following types of drifts:

- **Application policy drift** – This drift type identifies all the AppComponents that complied with the policy in the previous assessment but failed to comply in the current assessment.
- **Application resource drift** – This drift type identifies all the drifted resources in the current application version.

Resiliency assessment

AWS Resilience Hub uses a list of gaps and potential remedies to measure the effectiveness of a selected policy to recover and continue from a disaster. It evaluates each Application Component or application compliance status with the policy. This report includes cost optimization recommendations and references to potential issues.

Resiliency score

AWS Resilience Hub generates a score that indicates how closely your application follows our recommendations for meeting the application's resiliency policy, alarms, standard operating procedures (SOPs), and tests.

Disruption type

AWS Resilience Hub helps you assess resiliency against the following types of outages:

Application

The infrastructure is healthy, but the application or software stack doesn't operate as needed. This may occur after deployment of new code, configuration changes, data corruption, or malfunction of downstream dependencies.

Cloud Infrastructure

The cloud infrastructure is not functioning as expected because of an outage. An outage may occur because of a local error in one or more components. In most cases, this type of outage is resolved by rebooting, recycling, or reloading the faulty components.

Cloud Infrastructure AZ disruption

One or more Availability Zones are unavailable. This type of outage can be resolved by switching to a different Availability Zone.

Cloud Infrastructure Region incident

One or more Regions are unavailable. This type of incident can be resolved by switching to a different AWS Region.

AWS FIS experiments

AWS Resilience Hub recommends experiments using AWS FIS actions to verify application resiliency against different types of outages. These outages include application, infrastructure, Availability Zones (AZ), or AWS Region incidents of Application Components.

These experiments let you do the following:

- Inject a failure.
- Verify that alarms can detect an outage.
- Verify that recovery procedures, or standard operating procedures (SOPs), work correctly to recover the application from the outage.

Tests for SOPs measure estimated workload RTO and estimated workload RPO. You can test different application configurations and measure whether the output RTO and RPO meets the objectives defined in your policy.

SOP

A standard operating procedure (SOP) is a prescriptive set of steps that are designed to efficiently recover your application in the event of an outage or an alarm. Based on the application assessment, AWS Resilience Hub recommends a set of SOPs and it is recommended to prepare, test, and measure SOPs in advance of a disruption to ensure timely recovery.

AWS Resilience Hub personas

Building an enterprise application requires a collaborative effort from different cross-functional teams such as infrastructure, business continuity, application owner, and other stakeholders who are responsible for monitoring applications. The different personas from different teams contribute towards building and managing applications in AWS Resilience Hub, each having a different role and responsibilities. To learn more about granting permissions to different personas, see the section called “AWS Resilience Hub personas and IAM permissions reference”.

To get started with creating applications and running assessments in AWS Resilience Hub, we recommend you to create the following personas:

- **Infrastructure application manager** – Users with this persona are responsible for setting up, configuring, and maintaining infrastructure and application resources, ensuring the reliability and security of the application. Their responsibilities include the following:
  - Ensuring that the applications are deployed and updated regularly
  - Monitoring system performance
  - Troubleshooting issues
  - Implementing backup and disaster recovery plans
- **Business continuity manager** – Users with this persona are responsible for dictating application policies and determining the business criticality of applications. Their responsibilities include the following:
  - Taking key decisions in setting policies
  - Assessing business criticality
  - Allocating resources for critical applications
  - Assessing and managing risks
- **Application owner** – Users with this persona are responsible for ensuring highly available and reliable applications. Their responsibilities include the following:
  - Defining key performance identifiers for measuring and monitoring application performance and identifying bottlenecks
  - Organizing trainings for multiple stakeholders
  - Ensuring that the following documentation is up-to-date:
    - Application architecture
    - Deployment processes

    - Monitoring configurations
    - Performance optimization techniques
- **Read-only access** – Users with this persona are restricted to read-only permissions. Their responsibilities include maintaining visibility and oversight of an application's performance and health by monitoring resilience score, operational recommendations, and resiliency recommendations. In addition, they are also responsible for identifying issues, trends, and areas for improvement to ensure that the application meets the organization's objectives.

AWS Resilience Hub supported resources

Resources that affect application performance in the case of disruption are fully supported by AWS Resilience Hub top-level resources such as `AWS::RDS::DBInstance` and `AWS::RDS::DBCluster`.

To learn more about the permissions required for AWS Resilience Hub to include resources from all the supported services in your assessment, see the section called "AWSResilienceHubAsssessmentExecutionPolicy".

AWS Resilience Hub supports resources from the following AWS services:

- Compute
    - Amazon Elastic Compute Cloud (Amazon EC2)

      Note

      AWS Resilience Hub does not support the old Amazon Resource Name (ARN) format for accessing Amazon EC2 resources. The new ARN format uses your AWS account ID and enables the enhanced ability to tag resources in your cluster, and also tracks the cost of services and tasks running in your cluster.

      - **Old format (deprecated)** – `arn:aws:ec2:<region>::instance/<instance-id>`
      - **New format** – `arn:aws:ec2:<region>:<account-id>:instance/<instance-id>`

      For more information about the new ARN format, see Migrating your Amazon ECS deployment to the new ARN and resource ID format.
    - AWS Lambda

  - Amazon Elastic Kubernetes Service (Amazon EKS)
  - Amazon Elastic Container Service (Amazon ECS)
  - AWS Step Functions
- Database
  - Amazon Relational Database Service (Amazon RDS)
  - Amazon DynamoDB
  - Amazon DocumentDB
  - Amazon ElastiCache
- Networking and Content Delivery
  - Amazon Route 53
  - Elastic Load Balancing
  - Network Address Translation (NAT)
- Storage
  - Amazon Elastic Block Store (Amazon EBS)
  - Amazon Elastic File System (Amazon EFS)
  - Amazon Simple Storage Service (Amazon S3)
  - Amazon FSx for Windows File Server
- Others
  - Amazon API Gateway
  - Amazon Application Recovery Controller (ARC) (Amazon ARC)
  - Amazon Simple Notification Service
  - Amazon Simple Queue Service
  - AWS Auto Scaling
  - AWS Backup
  - AWS Elastic Disaster Recovery

Note

- AWS Resilience Hub provides additional transparency for your application resources by allowing you to view the supported instances of each resource. In addition, AWS Resilience Hub provides more accurate resiliency recommendations by identifying unique

instance of each resource while discovering the resource instances during assessment process. For more information about adding resource instances to your application, see Editing AWS Resilience Hub application resources.

- AWS Resilience Hub supports Amazon EKS and Amazon ECS on AWS Fargate.
- AWS Resilience Hub supports assessment of AWS Backup resource as a part of the following services:
  - Amazon EBS
  - Amazon EFS
  - Amazon S3
  - Amazon Aurora Global Database
  - Amazon DynamoDB
  - Amazon RDS services
  - Amazon FSx for Windows File Server
- Amazon ARC in AWS Resilience Hub assesses only Amazon DynamoDB global, Elastic Load Balancing, Amazon RDS, and AWS Auto Scaling groups.
- For AWS Resilience Hub to assess the cross-Region resources, group the resources under a single Application Component. For more information about the resources supported by each of the AWS Resilience Hub Application Components and grouping resources, see Grouping resources in an Application Component.
- Currently, AWS Resilience Hub does not support cross-Region assessments for Amazon EKS clusters if either the Amazon EKS cluster is located or if the application is created in an opt-in enabled AWS Region.
- Currently, AWS Resilience Hub assesses only the following Kubernetes resource types:
  - Deployments
  - ReplicaSets
  - Pods
- Currently, AWS Resilience Hub supports only the following engine types for ElastiCache resources:
  - Redis OSS engines

AWS Resilience Hub ignores the following types of resources:

- **Resources that do not affect estimated workload RTO or estimated workload RPO** – Resources such as `AWS::RDS::DBParameterGroup`, which does not affect estimated workload RTO or estimated workload RPO, is ignored by AWS Resilience Hub.
- **Non-top level resources** – AWS Resilience Hub only imports top-level resources, because they can derive other properties by querying the properties of top-level resources. For example, `AWS::ApiGateway::RestApi` and `AWS::ApiGatewayV2::Api` are supported resources for Amazon API Gateway. However, `AWS::ApiGatewayV2::Stage` is not a top-level resource. Therefore, it is not imported by AWS Resilience Hub.

Note

Unsupported resources

- You cannot identify multiple resources by using AWS Resource Groups (Amazon Route 53 RecordSets and API-GW HTTP) and Amazon Aurora Global resources. If you want to analyze these resources as part of your assessment, you must manually add the resource to the application. However, when you add Amazon Aurora Global resources for assessment, it must be grouped with the Amazon RDS instance's Application Component. For more information about editing resources, see the section called “Editing application resources”.
- These resources can affect application recovery, but they aren't fully supported by AWS Resilience Hub at this time. AWS Resilience Hub makes an effort to warn users about unsupported resources if the application is backed by an AWS CloudFormation stack, Terraform state file, AWS Resource Groups, or myApplications application.
- During the import process of an application's resources into AWS Resilience Hub, some resources may be ignored. When resources are ignored, it means they cannot be imported at all. However, resources marked as unsupported are currently not compatible with AWS Resilience Hub but may be supported in the future, allowing them to be included in the application for assessments. Additionally, AWS Resilience Hub might ignore certain resources if they are not supported by AWS Resource Groups. For more information about the resources supported by AWS Resource Groups, see Resource types you can use with AWS Resource Groups and Tag Editor.

AWS Resilience Hub and myApplications

The **Resiliency** widget in the myApplications dashboard streamlines the process of assessing and monitoring the application resilience. It enables you to quickly evaluate the resilience of your applications defined in myApplications without the need to manually recreate them in the AWS Resilience Hub console. This integrated approach combines the application management capabilities of myApplications with the resilience assessment features of AWS Resilience Hub, allowing you to leverage the strengths of both platforms. By bringing together application definitions and resilience assessment capabilities, the **Resiliency** widget simplifies the workflow, enabling you to access relevant information and take actions to enhance resilience from a centralized location. When an application is assessed from the **Resiliency** widget, AWS Resilience Hub performs the following:

- Creates the selected application in AWS Resilience Hub.
- Automatically discovers and maps the resources associated with the model.
- Creates and assigns a new resiliency policy with pre-defined values for recovery time objective (RTO) and recovery point objective (RPO). That is four hours for RTO and one hour for RPO. After you generate an assessment, you can modify the resiliency policy or assign a different policy from the AWS Resilience Hub console. For more information about updating resiliency policy and attaching a different policy, see Managing resiliency policies.
- Assesses the application's resilience against RTO and RPO defined in the resiliency policy to identify the areas that require improvements in the application architecture. The failure scenarios include Availability Zone failures, Regional outages, and other potential disruptions.
- Continuously monitors the application's resources and configuration changes after the initial assessment, providing alerts or updates if any changes impact the application's resilience.

> **Note**
>
> Before starting assessments, we recommend you to evaluate the potential costs involved in running assessments using AWS Resilience Hub. For detailed pricing information, see the AWS Resilience Hub pricing.

After assessing your application, you can access the full capability of AWS Resilience Hub from the widget by choosing Go to AWS Resilience Hub to view the application details in the AWS Resilience

Hub console. The process for including applications from myApplications into AWS Resilience Hub is governed by the following rules and constraints:

- You can associate only one myApplications application to an application in AWS Resilience Hub. That is, you can associate a myApplications application to an AWS Resilience Hub application either by running an assessment from **Resiliency** widget in the myApplications dashboard, or by completing the Using myApplications applications procedure while describing the application in AWS Resilience Hub console.
- You can only include, assess, and view myApplications applications that reside within the same AWS Region and AWS account boundaries as your myApplications environment. Applications created in different AWS Regions or under separate AWS accounts will not be visible or accessible through this widget.
- You can only add, remove, and update resources from the myApplications dashboard. When you modify the application resources from the myApplications dashboard, you must reimport the AWS Resilience Hub to view the resource changes in AWS Resilience Hub.

Learn more

For more information about managing applications and resources in the myApplications dashboard, see the following topics in AWS Console Home documentation:

- What is myApplications on AWS?
- Creating your first application in myApplications
- Managing resources
- Resiliency Widget

For more information about describing applications and running assessments in AWS Resilience Hub, see the following topics:

- To run a resiliency assessment for an existing **myApplications** application from **Resiliency** widget for the first time
- To rerun a resiliency assessment for an existing **myApplications** application from **Resiliency** widget
- Reviewing assessment summary in Resiliency widget

Getting started

This section describes how to start using AWS Resilience Hub. This includes creating AWS Identity and Access Management (IAM) permissions for an account.

Topics

- Prerequisites
- Add an application to AWS Resilience Hub

Prerequisites

Before you can use the AWS Resilience Hub, you must complete the following prerequisites:

- AWS accounts – Create one or more AWS accounts for each account type (primary/secondary/resource accounts) you want use within AWS Resilience Hub. For more information about creating and managing AWS accounts, see the following:
  - First time AWS user – Getting started: Are you a first-time AWS user?
  - Managing AWS account – https://docs.aws.amazon.com/accounts/latest/reference/managing-accounts.html
- AWS Identity and Access Management (IAM) permissions – After creating the AWS accounts, you must configure the required roles and IAM permissions for each of the accounts you have created. For example, if you have created an AWS account to access application resources, you must setup a new role and configure the necessary IAM permissions for AWS Resilience Hub to access the application resources from your account. To learn more about IAM permissions, see the section called “How AWS Resilience Hub works with IAM” and for more information about adding a policy to the role, see the section called “Defining trust policy using JSON file”.

  To get started quickly with adding IAM permissions to users, groups, and roles, you can use our AWS managed policies (the section called “AWS managed policies”). It is easier to use AWS managed policies to cover common use cases that are available in your AWS account than to write policies yourself. AWS Resilience Hub adds additional permissions to an AWS managed policy to extend support to other AWS services and to include new features. Hence:

  - If you are an existing customer and if you want your application to use the latest enhancements within your assessment, you must publish a new version of the application and then run a new assessment. For more information, see the following topics:
    - the section called “Publish a new application version”

- the section called “Running resiliency assessments in AWS Resilience Hub”
- If you are not using AWS managed policies to assign appropriate IAM permissions to users, groups, and roles, you must manually configure these permissions. For more information about AWS managed policies, see the section called “AWSResilienceHubAsssessmentExecutionPolicy”.

Add an application to AWS Resilience Hub

AWS Resilience Hub offers resiliency assessment and validation that integrates into your software development lifecycle. AWS Resilience Hub helps you proactively prepare and protect your AWS applications from disruptions by:

- Uncovering resiliency weaknesses.
- Estimating whether your target recovery time objective (RTO) and recovery point objective (RPO) can be met.
- Resolving issues before they are released into production.

This section guides you through adding an application. You gather resources from an existing myApplications application, AWS CloudFormation stacks, or AWS Resource Groups and create an appropriate resiliency policy. After describing an application, you can publish it in AWS Resilience Hub, and generate an assessment report on the resiliency of your application. You can then use recommendations from the assessment to improve resiliency. You can run another assessment, compare results, and then iterate until the estimated workload RTO and estimated workload RPO achieves your RTO and RPO targets.

Topics

- Get started by adding an application
- Select how this application is managed
- Add resource collections
- Set RTO and RPO
- Setup scheduled assessments and drift notification
- Setup permissions
- Configure the application configuration parameters
- Add tags

- Review and publish your AWS Resilience Hub application
- Run an assessment of your AWS Resilience Hub application

Get started by adding an application

Get started with AWS Resilience Hub by describing the details of your AWS application and running a report to assess resiliency.

To get started, on the AWS Resilience Hub home page under **Get started**, choose **Add application**.

To learn more about costs and billing associated with AWS Resilience Hub, see AWS Resilience Hub pricing.

Describe the details of your application in AWS Resilience Hub

This section shows you how to describe the details of your existing AWS application in AWS Resilience Hub.

To describe the details of your application

1. Enter a name for the application.
2. (Optional) Enter a description for the application.

Next

Select how this application is managed

Select how this application is managed

In addition to AWS CloudFormation stacks, AWS Resource Groups, myApplications applications, and Terraform state files, you can add resources that are located on Amazon Elastic Kubernetes Service (Amazon EKS) clusters. That is, AWS Resilience Hub allows you to add resources that are located on your Amazon EKS clusters as optional resources. This section provides the following options, which help you to determine the location of your application resources.

- **Resource collections** – Select this option if you want to discover resources from one of the resource collections. Resource collections include AWS CloudFormation stacks, AWS Resource Groups, myApplications applications, and Terraform state files.

If you select this option, you must complete one of the procedures in the section called "Add resource collections".

- **EKS only** – Select this option if you want to discover resources from namespaces within the Amazon EKS clusters.

  If you select this option, you must complete the procedure in the section called "Add EKS clusters"

- **Resource collections & EKS** – Select this option if you want to discover resources from AWS CloudFormation stacks, AWS Resource Groups, Terraform state files, and Amazon EKS clusters.

  If you select this option, complete one of the procedures in the section called "Add resource collections" and then complete the procedure in the section called "Add EKS clusters".

> **Note**
>
> For information about the number of resources supported per application, see Service Quotas.

Next

Add resource collections

Add resource collections

This section discusses the following options that you can use to form the basis of your application structure:

- Add resource collections
- Add EKS clusters

Add resource collections

This section discusses the following methods that you use to form the basis of your application structure:

- Using AWS CloudFormation stacks

- Using AWS Resource Groups
- Using myApplications applications
- Using Terraform state files

Using AWS CloudFormation stacks

Choose the AWS CloudFormation stacks that contain the resources you want to use in the application you're describing. The stacks can be from the AWS account that you are using to describe the application, or they can be from different accounts or different Regions.

To discover the resources that form the basis of your application structure

1. Select **CloudFormation stack** to discover your stack-based resources.
2. Choose stacks from the **Choose stacks** dropdown list that are associated with your AWS account and Region.

   To use stacks that are in a different AWS account, different Region, or both, choose the right arrow adjacent to **Add stack outside of AWS Region** and enter the Amazon Resource Name (ARN) of the stack in the **Enter a stack ARN** box, and then choose **Add stack ARN**. For more information about ARNs, see Amazon Resource Names (ARNs) in the *AWS General Reference*.

Using AWS Resource Groups

Choose the AWS Resource Groups that contain the resources that you want to use in the application that you're describing.

To discover the resources that form the basis of your application structure

1. Select **Resource groups** to discover the AWS Resource Groups that contain the resources.
2. Choose resources from **Choose a resource group** dropdown list.

   To use AWS Resource Groups that are in a different AWS account, different Region, or both, choose the right arrow adjacent to **Resource Group ARN:** and enter the Amazon Resource Name (ARN) of the AWS Resource Groups in the **Enter a resource group ARN** box, and then choose **Add resource Group ARN**. For more information about ARNs, see Amazon Resource Names (ARNs) in the *AWS General Reference*.

Using myApplications applications

Choose the myApplications application you want to include in AWS Resilience Hub

To include myApplications application in AWS Resilience Hub

1. Select **myApplications**.
2. Choose an application from the **Select application** dropdown list.

Using Terraform state files

Choose the Terraform state file that contains your Amazon S3 bucket resources that you want to use in the application you're describing. You can navigate to the location of your Terraform state file or provide a link to a Terraform state file you have access to that's located in a different Region.

> **Note**
>
> AWS Resilience Hub supports Terraform state file version `0.12` and later.

To discover the resources that form the basis of your application structure

1. Select **Terraform state files** to discover your S3 bucket resources.
2. From the **Select state files::** section, choose **Browse S3** to navigate to the location of your Terraform state file.

   To use Terraform state files located in a different Region, provide the link to the location of Terraform state file in the **S3 URI** field, and choose **Add S3 URL**.

   The limit for Terraform state files is 4 megabytes (MB).
3. From **Choose an archive in S3** dialog box, select your Amazon Simple Storage Service bucket from the **Buckets** section.
4. From the **Objects** section, select a key, and choose **Choose**.

Add EKS clusters

This section discusses about using Amazon EKS clusters to form the basis of your application structure.

> **Note**
>
> You must have Amazon EKS permissions and additional IAM roles to connect to the Amazon EKS cluster. For more information about adding single account and cross-account Amazon EKS permissions and additional IAM roles to connect to the cluster, see the following topics:
>
> - AWS Resilience Hub access permissions reference
> - the section called "Enabling AWS Resilience Hub access to your Amazon EKS cluster"

Choose the Amazon EKS clusters and namespaces that contain the resources you want to use in the application you're describing. The Amazon EKS clusters can be from the AWS account that you are using to describe the application, or they can be from different accounts or different Regions.

> **Note**
>
> For AWS Resilience Hub to assess your Amazon EKS clusters, you must manually add the relevant namespaces to each of the Amazon EKS clusters in **EKS clusters and namespaces** section. The namespace name must match exactly with the namespace name on your Amazon EKS clusters.

To add Amazon EKS clusters

1. In **1. Select EKS clusters** section, choose the Amazon EKS clusters from the **Choose EKS clusters** dropdown list that are associated with your AWS account and Region.
2. To use Amazon EKS clusters that are in a different AWS account, different Region, or both, choose the right arrow adjacent to **Add an EKS cluster within a different account or Region** and enter the Amazon Resource Name (ARN) of the Amazon EKS cluster in the **Enter an EKS ARN** box, and then choose **Add EKS ARN**. For more information about ARNs, see Amazon Resource Names (ARNs) in the *AWS General Reference*.

   For more information about adding permissions to access cross-Region Amazon Elastic Kubernetes Service clusters, see the section called "Enabling AWS Resilience Hub access to your Amazon EKS cluster".

To add namespaces from the selected Amazon EKS clusters

1. In the **Add namespaces** section, from the **EKS clusters and namespaces** table, select the radio button located at the left of Amazon EKS cluster name, and then choose **Update namespaces**.

   You can identify Amazon EKS clusters by the following:

   - **EKS cluster name** – Indicates the name of the selected Amazon EKS clusters.
   - **# of Namespaces** – Indicates the number of namespaces selected in the Amazon EKS clusters.
   - **Status** – Indicates whether AWS Resilience Hub has included the namespaces from the selected Amazon EKS clusters in your application. You can identify the status using the following options:
     - **Namespace required** – Indicates that you have not included any namespaces from the Amazon EKS cluster.
     - **Namespaces added** – Indicates that you have included one or more namespaces from the Amazon EKS cluster.
2. To add a namespace, in the **Update namespaces** dialog box, choose **Add a new namespace**.

   The **Update namespaces** dialog box displays all the namespaces that you have selected from your Amazon EKS cluster, as an editable option.
3. In the **Update namespaces** dialog box, you have the following edit options:

   - To add a new namespace, choose **Add a new namespace**, and then enter the namespace name in **namespace** box.

     The namespace name must exactly match with the namespace name on your Amazon EKS cluster.
   - To remove a namespace, choose **Remove** located next to the namespace.
   - To apply the selected namespaces to all the Amazon EKS clusters, choose **Apply namespaces to all EKS clusters**.

     If you choose this option, your previous namespace selection in the other Amazon EKS clusters will be overridden with the current namespace selection.
4. To include the updated namespaces in your application, choose **Update**.

Next

Set RTO and RPO

Set RTO and RPO

You can define a new resiliency policy with your own RTO/RPO targets, or you can choose an existing resiliency policy with predefined RTO/RPO targets. If you want to use one of the existing resiliency policies, select **Choose an existing policy** option and select an existing target application from the **Option item** drop-down list.

To define your own RTO/RPO targets

1. Select **Create a new resiliency policy** option.
2. Enter a name for the resiliency policy in the **Enter policy name** box (under **Name**).

   We have pre-populated this field with an auto-generated name. You can choose to use the same, or provide a different name.
3. (Optional) Enter a description for the resiliency policy in the **Description** box.
4. Define your RTO/RPO in the **RTO/RPO targets** section.

   > **Note**
   >
   > - We have pre-populated a default RTO and RPO for your application. You can change the RTO and RPO now, or after you assess the application.
   > - AWS Resilience Hub allows you to enter a value zero in the **RTO** and **RPO** fields of your resiliency policy. But, while assessing your application, the lowest possible assessment result is near zero. Hence, if you enter a value zero in the **RTO** and **RPO** fields, the estimated workload RTO and estimated workload RPO results will be near zero and the **Compliance status** for your application will be set to **Policy breached**.
5. To define RTO/RPO for your infrastructure and AZ, choose the right arrow to expand the **Infrastructure RTO and RPO** section.
6. In **RTO/RPO targets**, enter a numeric value in the box and then choose the unit of time that the value represents for both **RTO** and **RPO**.

   Repeat these entries for **Infrastructure** and **Availability Zone** in **Infrastructure RTO and RPO** section.

7. (Optional) If you have a multi-Region application and if you want to define a Region RTO and RPO, turn on **Region - Optional**.

   In **RTO** and **RPO**, enter a numeric value in the box and then choose the unit of time that the value represents for both **RTO** and **RPO**.

Next

the section called “Setup scheduled assessment and drift notification”

Setup scheduled assessments and drift notification

AWS Resilience Hub allows you to setup scheduled assessments and drift notification for assessing your application daily and getting notified when a drift is detected.

To setup drift notification

1. To assess your application daily, turn on **Automatically assess daily**.

   If this option is turned on, the daily assessment schedule begins only after the following:

   - The application is manually assessed successfully for the first time.
   - The application is configured with an appropriate IAM role.
   - If your application is configured with current IAM user permissions, you must create the `AWSResilienceHubAsssessmentExecutionPolicy`

     role using the appropriate procedure in the section called “How AWS Resilience Hub works with IAM”.
2. To get notified when AWS Resilience Hub detects any drifts from the resiliency policies, or when its resources have drifted, turn on **Get notified when the application drifts**.

   If this option is turned on, to receive drift notifications, you must specify an Amazon Simple Notification Service (Amazon SNS) topic. To provide Amazon SNS topic, in **Provide an SNS Topic** section, select **Choose an SNS topic** option and select an Amazon SNS topic from the **Choose an SNS topic** dropdown list.

> **Note**
>
> - To enable AWS Resilience Hub to publish notifications to your Amazon SNS topics, your Amazon SNS topic must be configured with appropriate permissions. For more information about configuring permissions, see the section called "Enabling AWS Resilience Hub to publish to your Amazon SNS topics".
> - Daily assessments can have an impact on your quota for runs. For more information about quotas, see AWS Resilience Hub endpoints and quotas in the *AWS General Reference*.

To use Amazon SNS topics that are in a different AWS account or different Region, or both, select **Enter SNS topic ARN** and enter the Amazon Resource Name (ARN) of the Amazon SNS topic in the **Provide an SNS topic** box. For more information about ARNs, see Amazon Resource Names (ARNs) in the *AWS General Reference*.

Next

Setup permissions

Setup permissions

AWS Resilience Hub allows you to configure the necessary permissions for **Primary account** and **Secondary account** to discover and assess the resources. However, you must run the procedure separately to configure permissions for each account.

To configure IAM roles and IAM permissions

1. To select an existing IAM role that will be used for accessing resources in the current account, select an IAM role from the **Select an IAM role** dropdown list.

> **Note**
>
> For a cross account setup, if you do not specify the Amazon Resource Names (ARNs) of the IAM role in the **Enter an IAM role ARN** box, AWS Resilience Hub will use the IAM role you have selected from the **Select an IAM role** dropdown list for all the accounts.

If there are no existing IAM roles attached to your account, you can create an IAM role by using one of the following options:

- **AWS IAM console** – If you choose this option, you must complete the procedure in **To create your AWS Resilience Hub role in the IAM console**.
- **AWS CLI** – If you choose this option, you must complete all the steps in **AWS CLI**.
- **CloudFormation template** – If you choose this option, depending on which account type (**Primary account** or **Secondary account**), you must create the roles using the appropriate AWS CloudFormation template.

2. Choose the right arrow to expand **Add IAM role(s) from a cross account - Optional** section.
3. To select IAM roles from a cross account, enter the ARNs of the IAM role in **Enter an IAM role ARN** box. Ensure that the ARNs of the IAM roles you are entering does not belong to the current account.
4. If you want to use current IAM user to discover your application resources, choose the right arrow to expand **Use the current IAM user permissions** section and select **I understand that I must manually configure permissions to enable the required functionality within AWS Resilience Hub**.

   If you select this option, some of the AWS Resilience Hub features (such as drift notification) may not function as expected and the inputs you have provided for creating a new application will be ignored.

Next

Configure the application configuration parameters

Configure the application configuration parameters

This section allows you to provide the details of your cross-Region failover support using AWS Elastic Disaster Recovery. AWS Resilience Hub will use this information to provide resiliency recommendations.

For more information about application configuration parameters, see Application configuration parameters.

To add application configuration parameters (Optional)

1. To expand the **Application configuration parameters** section, choose the right arrow.
2. Enter the failover account ID in the **Account ID** box. By default, we have pre-populated this field with your account ID that is used for AWS Resilience Hub, which can be changed.
3. Select a failover Region from the **Region** dropdown list.

> **Note**
>
> If you want to disable this feature, select "**–**" from the dropdown list.

Next

Add tags

Add tags

Assign a tag or label to an AWS resource to search and filter your resources, or track your AWS costs.

(Optional) To add tags to your application, choose **Add new tag** if you want to associate one or more tags with the application. For more information about tags, see Tagging resources in the *AWS General Reference*.

Choose **Add application** to create your application.

Next

Review and publish your AWS Resilience Hub application

Review and publish your AWS Resilience Hub application

After creating the application, you can still review the application and edit its resources. After you finish, choose **Publish** to publish the application.

> **Note**
>
> AWS Resilience Hub scans your application resources in the background and checks if they can be grouped in a more efficient way that will improve the accuracy of the assessments. If

AWS Resilience Hub identifies resources that can be grouped into relevant AppComponents, it displays **Resource grouping recommendations** information alert in the **Application structure** tab of the application page and you can review them by choosing **Review recommendations**. For more information, see the section called "AWS Resilience Hub resource grouping recommendations".

For more information about reviewing the application and editing its resources, see the following:

- the section called "Viewing application summary"
- the section called "Editing application resources"

Next

Run an assessment of your AWS Resilience Hub application

Run an assessment of your AWS Resilience Hub application

The application that you published is listed on the **Summary** page.

After you publish your AWS Resilience Hub application, you are redirected to the application summary page where you can run a resiliency assessment. The assessment evaluates your application configuration against the resiliency policy that is attached to your application. An assessment report is generated that shows how your application measures against the objectives in your resiliency policy.

To run a resiliency assessment

1. On the **Applications summary** page, choose **Assess resiliency**.
2. In the **Run resiliency assessment** dialog, enter a unique name for the report or use the generated name in the **Report name** box.
3. Choose **Run**.
4. After you are notified that the assessment report has been generated, choose the **Assessments** tab and your assessment to view the report.
5. Choose the **Review** tab to view your application's assessment report.

Using AWS Resilience Hub

AWS Resilience Hub helps you improve the resiliency of your applications on AWS and reduce the recovery time in the event of application outages.

Topics:

- AWS Resilience Hub summary
- AWS Resilience Hub dashboard
- Describing and managing AWS Resilience Hub Applications
- Managing resiliency policies
- Running and managing resiliency assessments in AWS Resilience Hub
- Running and managing resiliency assessments from Resiliency widget
- Managing alarms
- Managing standard operating procedures
- Managing AWS Fault Injection Service experiments
- Understanding resiliency scores
- Integrating operational recommendations into your application with CloudFormation

AWS Resilience Hub summary

AWS Resilience Hub provides a visual summary with charts and graphs that gives you an at-a-glance view of your application's resilience posture across multiple AWS services and resources. This comprehensive and concise visual summary enables you to quickly identify potential resilience gaps, prioritize actions, and track progress in enhancing your application's ability to recover from disruptions. When you choose **Export**, and if you are exporting the metrics for the first time, AWS Resilience Hub creates a new Amazon S3 bucket in the Region from which you are accessing AWS Resilience Hub. This Amazon S3 bucket is created only for the first time and will be used to save the exported metrics upon successful completion. Additional charges apply for storing exported data in Amazon S3. For more information about these charges, Amazon S3 pricing.

The charts and graphs in the widgets help you understand the following:

- Overview of the application's overall resilience score and current operational state.

- Potential policy violations or deviations from best practices by highlighting applications that are not compliant with established policies or have drifted from recommended configurations. Additionally, it also highlights specific areas that enables you to prioritize and address them.
- Critical resources or applications that demand immediate attention.
- Recommendations for enhancing resilience practices, such as implementing alarms, conducting AWS Fault Injection Service (AWS FIS) experiments, and establishing standard operating procedures. These recommendations are tracked over time, allowing you to monitor the implementation progress and measure the impact on the application's overall resilience posture.

Widgets

- Application status
- Top infrastructure recommendations by resource type
- Infrastructure recommendations
- Unimplemented operational recommendations
- Alarm recommendations
- SOP recommendations
- AWS FIS experiment recommendations
- Applications with drifts
- Resiliency score
- Bottom 10 applications for resiliency score
- Application state by policy

Application status

This widget indicates if your applications comply with the resiliency policy or not. Choose the number adjacent to the **Application count** in the pop-up to view all the associated applications in the **Applications** pane. To view all the applications you have created, choose **View applications**. For more information about managing applications in AWS Resilience Hub, see Viewing an AWS Resilience Hub application summary.

Top infrastructure recommendations by resource type

This widget displays the number of infrastructure recommendations for each resource type of your AWS resources provided in the last successful assessment to improve their resiliency

posture. You can identify the details by hovering over them or by navigating to them. To view all the applications you have created, choose **View applications**. For more information about infrastructure recommendations, see Reviewing resiliency recommendations.

Infrastructure recommendations

This widget lists up to 10 applications that have the maximum number of infrastructure recommendations provided in the last successful assessment to improve their resiliency posture. To view all the applications you have created, choose **View applications**. For more information about infrastructure recommendations, see Reviewing resiliency recommendations.

You can identify the details using the following:

- **Application name** – Name of the application that you provided while defining it in AWS Resilience Hub.
- **Count** – Indicates the number of infrastructure recommendations provided by AWS Resilience Hub in the last successful assessment. Choose the number to view all the infrastructure recommendations provided in the assessment report.
- **Last assessed** – Indicates the date and time when your application was last assessed successfully.

Unimplemented operational recommendations

This widget lists up to 10 applications that have the maximum number of unimplemented operational recommendations provided in the last successful assessment to improve their resiliency posture. To view all the applications you have created, choose **View applications**. For more information about operational recommendations, see Reviewing operational recommendations.

You can identify the details using the following:

- **Application name** – Name of the application that you provided while defining it in AWS Resilience Hub.
- **Count** – Indicates the number of operational recommendations provided by AWS Resilience Hub in the last successful assessment. Choose the number to view all the unimplemented operational recommendations in the assessment report.
- **Last assessment time** – Indicates the date and time when your application was last assessed successfully.

Alarm recommendations

This widget lists all the Amazon CloudWatch alarm recommendations provided for improving the resilience posture over a selected time period. The different categories (**Implemented**, **Not implemented**, and **Excluded**) indicate their implementation state in your application. You can view the number of Amazon CloudWatch alarm recommendations for each category by hovering over them or by navigating to them. To view all the applications you have created, choose **View applications**. For more information about alarm recommendations, see Reviewing operational recommendations.

SOP recommendations

This widget lists all the standard operating procedure (SOP) recommendations provided for improving the resilience posture over a selected time period. The different categories (**Implemented**, **Not implemented**, and **Excluded**) indicate their implementation state in your application. You can view the number of SOP recommendations for each category by hovering over them or by navigating to them. To view all the applications you have created, choose **View applications**. For more information about operational recommendations, see Reviewing operational recommendations.

AWS FIS experiment recommendations

This widget lists all the AWS FIS experiment recommendations provided for improving the resilience posture over a selected time period. The different categories (**Implemented**, **Not implemented**, **Partially implemented**, and **Excluded**) indicate their implementation state in your application. You can view the number of AWS FIS experiment recommendations for each category by hovering over them or by navigating to them. To view all the applications you have created, choose **View applications**. For more information about AWS FIS experiment recommendations, see Managing standard operating procedures.

Applications with drifts

This widget lists all your applications that have drifted from their previous compliant state in the last successful assessment. To view all the applications you have created, choose **View applications**. For more information about managing applications in AWS Resilience Hub, see Viewing an AWS Resilience Hub application summary.

You can identify the details using the following:

- **Application name** – Name of the application that you provided while defining it in AWS Resilience Hub.
- **Policy drifts** – Choose the number adjacent to the application name to view all the Application Components that complied with the policy in the previous assessment but failed to comply in the current assessment.
- **Resource drifts** – Choose the number below to view all the resources that have changed from their configuration in the latest import.

Resiliency score

This widget displays the trend of the application's resiliency score over a selected time period for up to five applications. You can view an application's resiliency score by hovering over the line associated with the application name or by navigating to it, and then choosing the application name to view the application summary. To view all the applications you have created, choose **View applications**. For more information about resilience score, see Understanding resiliency scores.

Bottom 10 applications for resiliency score

This widget lists up to 10 applications with the lowest resiliency scores from their most recent assessments, highlighting the applications that require immediate attention to improve their resilience. To view all the applications you have created, choose **View applications**. For more information about resilience score, see Understanding resiliency scores.

You can identify the details using the following:

- **Application name** – Name of the application that you provided while defining it in AWS Resilience Hub.
- **Resiliency score** – The overall resiliency score determined by AWS Resilience Hub for your application after running the assessment.
- **Last assessment time** – Indicates the date and time when your application was last assessed successfully.

Application state by policy

This widget lists all your policies and the number of applications that have breached, met, or yet to be assessed against them. To view all the policies you have created, choose **View policies**. For more information about resilience score, see Managing resiliency policies.

You can identify the details using the following:

- **Policy name** – Indicates the policy name you provided while defining it in AWS Resilience Hub.
- **Type** – Indicates that the type of policy (**Resiliency policy**) attached to the application.
- **Policy name** – Indicates the number of applications that have either breached the RTO and RPO targets defined in the resiliency policy.
- **Apps met** – Indicates the number of applications that are compliant with the resiliency policy.
- **Apps not assessed** – Indicates the number of applications that are yet to be assessed against the resiliency policy.
- **Resiliency score** – The overall resiliency score determined by AWS Resilience Hub for your application after running the assessment.
- **Last assessment time** – Indicates the date and time when your application was last assessed successfully.

AWS Resilience Hub dashboard

The dashboard provides a comprehensive view of the resilience status of your application portfolio. The dashboard aggregates and organizes resilience events (for example, unavailable database or failed resilience validation), alerts, and insights from services such as CloudWatch and AWS Fault Injection Service (AWS FIS).

The dashboard also generates a resilience score for each application that's assessed. This score indicates how well your application performs when assessed against recommended resilience policies, alarms, recovery standard operating procedures (SOPs), and tests. You can use this score to measure resilience improvements over time.

To view AWS Resilience Hub dashboard, choose **Dashboard** from navigation menu. The **Dashboard** page displays the following sections:

Application status

The application statuses indicate whether the applications have been assessed for compliance with their attached resiliency policy or not. In addition, after an assessment is completed, the status also indicates if the input sources of your applications have been modified or not. Choose a number under each of the following statuses to view all the applications that share the same status in the **Applications** page:

- **Applications in policy** – Indicates all the applications that comply with their attached resiliency policy.
- **Applications breaching policy** – Indicates all the applications that does not comply with their attached resiliency policy.
- **Applications not assessed** – Indicates all the applications whose compliance has not been assessed or tracked yet.
- **Applications drifted** – Indicates all the applications that have drifted from their resiliency policy or if their resources have drifted.

Application resiliency score over time

With the application resiliency score over time, you can view a graph of your application's resiliency over the past 30 days. While the dropdown menu can list 10 of your applications, AWS Resilience Hub only shows you a graph of up to four applications at a time. For more information about resiliency score, see Understanding resiliency scores.

> **Note**
>
> AWS Resilience Hub does not run scheduled assessments at the same time. As a result, you may need to return to the resiliency score over time graph at a later time to view the daily assessment of your applications.

AWS Resilience Hub also uses Amazon CloudWatch to generate these graphs. Choose **View metrics in CloudWatch** to create and view more granular information about your application's resiliency in your CloudWatch dashboard. For more information about CloudWatch, see Using dashboards in the *Amazon CloudWatch User Guide*.

Implemented alarms

This section lists all the alarms that you have set up in Amazon CloudWatch to monitor all the applications. For more information , see Viewing alarms.

Implemented experiments

This section lists all fault injection experiments that you have implemented in all the applications. For more information, see Viewing AWS FIS experiments.

Describing and managing AWS Resilience Hub Applications

An AWS Resilience Hub application is a collection of AWS resources that are structured to prevent and recover AWS application disruptions.

To describe an AWS Resilience Hub application, you provide an application name, resources from one or more CloudFormation stacks, and an appropriate resiliency policy. You can also use any existing AWS Resilience Hub application as a template to describe your application.

After you describe an AWS Resilience Hub application, you must publish it so that you can run a resiliency assessment on it. You can then use recommendations from the assessment to improve resiliency by running another assessment, comparing results, and then reiterating the process until your estimated workload RTO and estimated workload RPO meet your RTO and RPO targets.

To view the **Applications** page, choose **Applications** from the navigation pane. You can identify your applications in the **Applications** page by the following:

- **Name** – The name of the application you had provided while defining it in AWS Resilience Hub.
- **Description** – The description of the application you had provided while defining it in AWS Resilience Hub.
- **Compliance status** – AWS Resilience Hub sets the application status as **Assessed**, **Not assessed**, **Policy breached**, or is **Changes detected**.
  - **Assessed** - AWS Resilience Hub has assessed your application.
  - **Not assessed** - AWS Resilience Hub has not assessed your application.
  - **Policy breached** - AWS Resilience Hub has determined your application did not meet your resiliency policy's objectives for Recovery Time Objective (RTO) and Recovery Point Objective (RPO). Review and use the recommendations provided by AWS Resilience Hub before reassessing your application for resiliency. For more information about recommendations, see Add an application to AWS Resilience Hub.
  - **Changes detected** - AWS Resilience Hub has detected changes made to the resiliency policy associated with your application. You must reassess your application for AWS Resilience Hub to determine if your application meets your resiliency policy's objectives.
- **Scheduled assessments** – The resource type identifies the component resource for your application. For more information about scheduled assessments, see Application resiliency.
  - **Active** - This indicates your application is automatically assessed daily by AWS Resilience Hub.
  - **Disabled** - This indicates your application is not automatically assessed daily by AWS Resilience Hub and you must manually assess your application.

- **Drift status** – Indicates if your application has drifted or not from the previous successful assessment and sets one of the following statuses:
  - **Drifted** - Indicates that the application, which was compliant with its resiliency policy in the previous successful assessment, has now breached the resiliency policy and the application is at risk. Additionally, it also indicates if the resources within input sources, which are included in the current application version, were added or removed.
  - **Not drifted** - Indicates that the application is still estimated to meet its RTO and RPO targets defined in the policy. Additionally, it also indicates that the resources within input sources, which are included in the current application version, were not added or removed.
- **Estimated workload RTO** – Indicates the maximum possible estimated workload RTO of your application. This value is the maximum estimated workload RTO of all the disruption types from the last successful assessment.
- **Estimated workload RPO** – Indicates the maximum possible estimated workload RPO of your application. This value is the maximum estimated workload RTO of all the disruption types from the last successful assessment.
- **Last assessment time** – Indicates the date and time your application was last assessed successfully.
- **Creation time** – The date and time that the application was created.
- **ARN** – The Amazon Resource Name (ARN) of your application. For more information about ARNs, see Amazon Resource Names (ARNs) in the *AWS General Reference*.

> **Note**
>
> AWS Resilience Hub can fully assess the resiliency of cross-Region Amazon ECS resources only if you are using Amazon ECR for the image repository.

In addition, you can also filter the applications list by using one of the following options in the **Applications** page:

- **Find applications** – Enter your application name to filter the results by the name of your application.
- **Filter last assessment time by a date and time range** – To apply this filter, choose the calendar icon and select one of the following options to filter by the results that matches the time range:
  - **Relative range** – Select one of the available options and choose **Apply**.

If you choose **Customised range** option, enter a duration in **Enter duration** box and select the appropriate unit of time from **Unit of time** dropdown list, then choose **Apply**.

- **Absolute range** – To specify the date and time range, provide the start time and end time, and then choose **Apply**.

The following topics show the different approaches for describing an AWS Resilience Hub application and how to manage them.

Topics

- Viewing an AWS Resilience Hub application summary
- Editing AWS Resilience Hub application resources
- Managing Application Components
- Publishing a new AWS Resilience Hub application version
- Viewing all the AWS Resilience Hub application versions
- Viewing resources of AWS Resilience Hub application
- Deleting an AWS Resilience Hub application
- Application configuration parameters

Viewing an AWS Resilience Hub application summary

The application summary page in the AWS Resilience Hub console provides an overview of your application information and resiliency health.

To view an application summary

1. Choose **Applications** from the navigation pane.
2. On the **Applications** page, choose the name of the application you want to view.

The applications summary page has the following sections.

Topics

- Assessment Summary
- Summary

- Application resiliency
- Implemented alarms
- Implemented experiments

Assessment Summary

This section provides a summary of the last successful assessment and highlights critical recommendations as actionable insights. AWS Resilience Hub uses Amazon Bedrock generative AI capabilities to help focus users on the most critical resilience recommendations provided by AWS Resilience Hub. By focusing on the critical items, you can focus on the most critical recommendations that improves the resilience posture of your application. Choose a recommendation to view its summary and choose **View details** to view more details about the recommendations in the relevant section of the assessment report. For more information about reviewing the assessment report, see the section called “Reviewing assessments reports”.

> **Note**
>
> - This assessment summary is available only in US East (N. Virginia) Region.
> - The assessment summary generated by large language models (LLMs) on Amazon Bedrock are only suggestions. The current level of generative AI technology is not perfect and LLMs are not infallible. Bias and incorrect answers, although rare, should be expected. Review each recommendation in the **Assessment summary** before you use the output from an LLM.

Summary

This section provides a summary of the selected application in the following sections:

- **Application info** – This section provides the following information about the selected application:
  - **Application status** – Indicates the status of the application.
  - **Description** – The description of the application.
  - **Version** – Indicates the currently assessed version of the application.
  - **Resiliency policy** – Indicates the resiliency policy that is attached the application. For more information about resiliency policies, see Managing resiliency policies.

- **Application drifts** – This section highlights the drifts detected while running an assessment for the selected application to check if it is compliant with its resiliency policy. Additionally, it also checks if any of the resources have been added or removed since the last time the application version was published. This section displays the following information:
  - **Policy drifts** – Choose the number below to view all the Application Components that complied with the policy in the previous assessment but failed to comply in the current assessment.
  - **Resource drifts** – Choose the number below to view all the drifted resources in the latest assessment.

Application resiliency

The metrics shown in the **Resiliency score** section are from the most recent resiliency assessment of the application.

Resiliency score

The resiliency score helps you quantify your readiness to handle a potential disruption. This score reflects how closely your application has followed the AWS Resilience Hub recommendations for meeting the application's resiliency policy, alarms, standard operating procedures (SOPs), and tests.

The maximum resiliency score that your application can achieve is 100%. The score represents all recommended tests that run in a predefined period of time. It indicates that the tests are initiating the correct alarm, and that the alarm initiates the correct SOP.

For example, suppose that AWS Resilience Hub recommends one test with one alarm and one SOP. When the test runs, the alarm initiates the associated SOP, and then runs successfully. For more information about the resiliency score, see Understanding resiliency scores.

Implemented alarms

The application summary **Implemented alarms** section lists the alarms that you set up in Amazon CloudWatch to monitor the application. For more information about alarms, see Managing alarms.

Implemented experiments

The application summary **Fault injection experiments** section shows a list of the fault injection experiments. For more information about fault injection experiments, see Managing AWS Fault Injection Service experiments.

Editing AWS Resilience Hub application resources

To receive accurate and helpful resiliency assessments, ensure that your application description is updated and matches your actual AWS application and resources. Assessment reports, validation, and recommendations are based on the listed resources. If you add or remove resources from an AWS application, you should reflect those changes in AWS Resilience Hub.

AWS Resilience Hub provides transparency about your application sources. You can identify and edit the resources and the application sources in your application.

> **Note**
>
> Editing the resources modifies only the AWS Resilience Hub reference of your application. No changes are made to your actual resources.

You can add resources that are missing, modify existing resources, or remove resources that you don't need. Resources are grouped into logical Application Components (AppComponents). You can edit the AppComponents to better reflect the structure of your application.

Add to or update your application resources by editing a draft version of your application and publishing the changes to a new (release) version. AWS Resilience Hub uses the release version (which includes the updated resources) of your application for running resiliency assessments.

To assess the resiliency of your application

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the application name that you want to edit.
3. From **Actions** menu, choose **Assess resiliency**.
4. In the **Run resiliency assessment** dialog, enter a unique name for the report or use the generated name in the **Report name** box.
5. Choose **Run**.
6. After you are notified that the assessment report has been generated, choose the **Assessments** tab and your assessment to view the report.
7. Choose the **Review** tab to view your application's assessment report.

To enable scheduled assessment

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, select the application for which you want to enable scheduled assessment.
3. Turn on **Automatically assess daily**.

To disable scheduled assessment

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, select the application for which you want to enable scheduled assessment.
3. Turn off **Automatically assess daily**.

> **Note**
>
> Disabling scheduled assessment will disable drift notification.

4. Choose **Turn off**.

To enable drift notification for your application

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, select the application for which you want to enable drift notification or edit the drift notification settings.
3. You can edit drift notification by choosing one of the following options:

   - From **Actions**, choose **Enable drift notification**.
   - Choose **Enable notification** in **Application drifts** section.

4. Complete the steps in Setup scheduled assessments and drift notification, and then return to this procedure.
5. Choose **Enable**.

   Enabling drift notification will also enable scheduled assessment.

To edit drift notification for your application

> **Note**
>
> This procedure is applicable if you have enabled scheduled assessment (**Automatically assess daily** is turned on) and drift notification.

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, select the application for which you want to enable drift notification or edit the drift notification settings.
3. You can edit drift notification by choosing one of the following options:
   - From **Actions**, choose **Edit drift notification**.
   - Choose **Edit notification** in **Application drifts** section.
4. Complete the steps in Setup scheduled assessments and drift notification, and then return to this procedure.
5. Choose **Save**.

To update the security permissions of your application

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, select the application for which you want to update the security permissions.
3. From **Actions**, choose **Update permissions**.
4. To update the security permissions, complete the steps in Setup permissions, and then return to this procedure.
5. Choose **Save and update**.

To attach a resiliency policy to your application

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the application name that you want to edit.
3. From **Actions** menu, choose **Attach resiliency policy**.

4. In the **Attach policy** dialog, select a resiliency policy from **Select a resiliency policy** dropdown list.
5. Choose **Attach**.

To edit input sources, resources, and AppComponents of your application

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the application name that you want to edit.
3. Choose the **Application structure** tab.
4. Choose the plus sign **+** before **Version**, and then select the application version with **Draft** status.
5. To edit input sources, resources, and AppComponents of your application, complete the steps in the following procedures.

To edit the input sources of your application

1. To edit the input sources of your application, choose the **Input sources** tab.

   The **Input sources** section lists all the input sources of your application resources. You can identify the input sources by the following:

   - **Source name** – The name of the input source. Choose a source name to view its details in the respective application. For manually added input sources, the link will not be available. For example, if you choose the source name that is imported from an AWS CloudFormation stack, you will be redirected to the stack details page on the AWS CloudFormation console.
   - **Source ARN** – The Amazon Resource Name (ARN) of the input source. Choose an ARN to view its details in the respective application. For manually added input sources, the link will not be available. For example, if you choose an ARN that is imported from an AWS CloudFormation stack, you will be redirected to the stack details page on the AWS CloudFormation console.
   - **Source Type** – The type of input source. Input sources include Amazon EKS clusters, AWS CloudFormation stacks, myApplications applications, AWS Resource Groups, Terraform state files, and manually added resources.
   - **Associated resources** – The number of resources that are associated with the input source. Choose a number to view all the associated resources of an input source in the **Resources** tab.

2. To add input sources to your application, from the **Input sources** section, choose **Add input sources**. For more information about adding input sources, see the section called "Add resources to your AWS Resilience Hub application".
3. To edit input sources, select the input sources and choose one the following options from **Actions**:

   - **Reimport input sources (up to 5)** – Reimports up to five selected input sources.
   - **Delete input sources** – Deletes the selected input sources.

     To publish an application, it must contain a minimum of one input source. If you delete all the input sources, **Publish new version** will be disabled.

To edit the resources of your application

1. To edit the resources of your application, choose the **Resources** tab.

   > **Note**
   >
   > To see the list of unassessed resources, choose **View unassessed resources**.

   The **Resources** section lists resources from the application that you chose to use as a template for your application description. To enhance your search experience, AWS Resilience Hub has grouped resources based on multiple search criteria. These search criteria include AppComponent types, **Unsupported** resources, and **Excluded** resources. To filter the resources based on a search criteria in the **Resources** table, choose the number below each of the search criteria.

   You can identify the resources by the following:

   - **Logical ID** – A logical ID is a name used to identify resources in your AWS CloudFormation stack, Terraform state file, manually added application, myApplications application, or AWS Resource Groups.

> **Note**
>
> - Terraform lets you use the same name for different resource types. Therefore, you see "- *resource type*" at the end of the logical ID for resources that share the same name.
> - To view the instances of all the application resources, choose the plus (**+**) sign before the **Logical ID**. To view all the instances of an application resource, choose the plus (**+**) sign before the Logical ID of each resource.
>
> For more information about the supported resources, see the section called "Supported AWS Resilience Hub resources".

- **Resource type** – The resource type identifies the component resource for your application. For example, `AWS::EC2::Instance` declares an Amazon EC2 instance. For more information about grouping AppComponent resources, see Grouping resources in an Application Component.
- **Source name** – The name of the input source. Choose a source name to view its details in the respective application. For manually added input sources, the link will not be available. For example, if you choose the source name that is imported from an AWS CloudFormation stack, you will be redirected to the stack details page on the AWS CloudFormation.
- **Source Type** – The type of input source. Input sources include AWS CloudFormation stacks, myApplications applications, AWS Resource Groups, Terraform state files, and manually added resources.

> **Note**
>
> To edit your Amazon EKS clusters, complete the steps in **To edit the input sources of your AWS Resilience Hub application** procedure.

- **Source stack** – The AWS CloudFormation stack that contains the resource. This column depends on the type of application structure that you selected.
- **Physical ID** – The actual assigned identifier for that resource, such as an Amazon EC2 instance ID or an S3 bucket name.
- **Included** – This indicates whether AWS Resilience Hub includes these resources in the application.

- **Assessable** – This indicates whether the AWS Resilience Hub will assess your resource for resiliency.
- **AppComponents** – The AWS Resilience Hub component that was assigned to this resource when its application structure was discovered.
- **Name** – Name of the application resource.
- **Account** – The AWS account that owns the physical resource.

2. To find a resource that is not listed, enter the resource logical ID in the search box.
3. To remove a resource from your application, select the resource, and then choose **Exclude resource** from **Actions**.
4. To resolve the resources on your application, choose **Refresh resources**.
5. To modify your existing application resources, complete the following steps:

   a. Select a resource, and then choose **Update stacks** from **Actions**.

   b. In the **Update stacks** page, to update your resources, complete the appropriate procedures in Add resource collections, and then return to this procedure.

   c. Choose **Save**.

6. To add a resource to your application, from **Actions**, choose **Add resource** and complete the following steps:

   a. Select a resource type from the **Resource type** dropdown list.

   b. Select an AppComponent from the **AppComponent** dropdown list.

   c. Enter the resource logical ID in the **Resource name** box.

   d. Enter the physical resource ID, or resource name, or the resource ARN in the **Resource identifier** box.

   e. Choose **Add**.

7. To edit the resource name, select a resource, choose **Edit resource name** from **Actions**, and then complete the following steps:

   a. Enter the resource logical ID in the **Resource name** box.

   b. Choose **Save**.

8. To edit the resource identifier, select a resource, choose **Edit resource identifier** from **Actions**, and then complete the following steps:

a. Enter the physical resource ID, or resource name, or the resource ARN in the **Resource identifier** box.

b. Choose **Save**.

9. To change the AppComponent, select a resource, choose **Change AppComponent** from **Actions**, and complete the following steps:

a. Select an AppComponent from the **AppComponent** dropdown list.

b. Choose **Add**.

10. To delete a resource, select a resource, and then choose **Delete resource** from **Actions**.

11. To include a resource, select a resource, and then choose **Include resource** from **Actions**.

To edit the AppComponents of your application

1. To edit the AppComponents of your application, choose the **AppComponents** tab.

> **Note**
>
> For more information about grouping AppComponent resources, see Grouping resources in an Application Component.

The **AppComponents** section lists all the logical components that the resources are grouped into. You can identify the AppComponents by the following:

- **AppComponent name** – The name of the AWS Resilience Hub component that was assigned to this resource when its application structure was discovered.
- **AppComponent type** – The type of AWS Resilience Hub component.
- **Source name** – The name of the input source. Choose a source name to view its details in the respective application. For example, if you choose the source name that is imported from an AWS CloudFormation stack, you will be redirected to the stack details page on the AWS CloudFormation.
- **Resource count** – The number of resources that are associated with the input source. Choose a number to view all the associated resources of an input source in the **Resources** tab.

2. To create an AppComponent, from **Actions** menu, choose **Create new AppComponent** and complete the following steps:

a. Enter a name for the AppComponent in the **AppComponent name** box. For reference, we have pre-populated this field with a sample name.

b. Select the type of AppComponent from the **AppComponent type** dropdown list.

c. Choose **Save**.

3. To edit an AppComponent, select an AppComponent, and then choose **Edit AppComponent** from **Actions**.
4. To delete an AppComponent, select an AppComponent, and then choose **Delete AppComponent** from **Actions**.

After you make changes to your resource list, you will receive an alert indicating that changes have been made to the draft version of your application. To run an accurate resiliency assessment, you must publish a new version of your application. For more information about how to publish a new version, see Publishing a new AWS Resilience Hub application version.

Managing Application Components

An Application Component (AppComponent) is a group of related AWS resources that work and fail as a single unit. For example, if you have a primary and replica database, both the databases belong to the same AppComponent. AWS Resilience Hub has rules that govern which AWS resources can belong to which AppComponent type. For example, a `DBInstance` can belong to `AWS::ResilienceHub::DatabaseAppComponent` and not to `AWS::ResilienceHub::ComputeAppComponent`.

The AWS Resilience Hub AppComponents support the following resources:

- `AWS::ResilienceHub::ComputeAppComponent`
  - `AWS::ApiGateway::RestApi`
  - `AWS::ApiGatewayV2::Api`
  - `AWS::AutoScaling::AutoScalingGroup`
  - `AWS::EC2::Instance`
  - `AWS::ECS::Service`
  - `AWS::EKS::Deployment`
  - `AWS::EKS::ReplicaSet`
  - `AWS::EKS::Pod`

  - AWS::Lambda::Function
  - AWS::StepFunctions::StateMachine
- AWS::ResilienceHub::DatabaseAppComponent
  - AWS::DocDB::DBCluster
  - AWS::DynamoDB::Table
  - AWS::ElastiCache::CacheCluster
  - AWS::ElastiCache::GlobalReplicationGroup
  - AWS::ElastiCache::ReplicationGroup
  - AWS::ElastiCache::ServerlessCache
  - AWS::RDS::DBCluster
  - AWS::RDS::DBInstance
- AWS::ResilienceHub::NetworkingAppComponent
  - AWS::EC2::NatGateway
  - AWS::ElasticLoadBalancing::LoadBalancer
  - AWS::ElasticLoadBalancingV2::LoadBalancer
  - AWS::Route53::RecordSet
- AWS:ResilienceHub::NotificationAppComponent
  - AWS::SNS::Topic
- AWS::ResilienceHub::QueueAppComponent
  - AWS::SQS::Queue
- AWS::ResilienceHub::StorageAppComponent
  - AWS::Backup::BackupPlan
  - AWS::EC2::Volume
  - AWS::EFS::FileSystem
  - AWS::FSx::FileSystem

> **Note**
>
> Currently, AWS Resilience Hub supports Amazon FSx for Windows File Server only.

  - AWS::S3::Bucket

Topics

- Grouping resources in an Application Component

Grouping resources in an Application Component

When the application is imported into AWS Resilience Hub along with its resources, AWS Resilience Hub makes its best effort to group related resources into the same AppComponent when you import your application, but the grouping might not always be 100 percent accurate. Some resources are blocked for manual grouping and will be grouped automatically when applicable because these services have strict dependencies that require specific grouping configurations. For a complete list of services that are blocked for manual grouping, see the section called "Blocked services for manual grouping".

AWS Resilience Hub performs the following activities after your application and its resources are successfully imported:

- Scans your resources to check if they can be re-grouped into new AppComponents to improve the assessment accuracy.
- If AWS Resilience Hub identifies resources that can be re-grouped into new AppComponents, it displays the same as recommendations and allows you to either accept or reject the same. In AWS Resilience Hub, the confidence level assigned to a grouping recommendation indicates the degree of certainty with which the resources should be grouped together based on their attributes and metadata. A **High** confidence level indicates that AWS Resilience Hub has a confidence level of 90% or above that the resources in that group are related and should be grouped together. A **Medium** confidence level indicates that AWS Resilience Hub has a confidence level between 70% and 90% that the resources in that group are related and should be grouped together.

> **Note**
>
> AWS Resilience Hub requires the correct grouping so that it can compute estimated workload RTO and estimated workload RPO to generate recommendations.

The following are examples of correct groupings:

- Group primary databases and replicas under a single AppComponent.

- Group Amazon EC2 instances that run the same application under a single AppComponent.
- Group Amazon ECS services in one Region and failover Amazon ECS services in another Region under a single AppComponent.

For more information about reviewing and including resource grouping recommendations by AWS Resilience Hub, see the following topics:

- AWS Resilience Hub resource grouping recommendations
- Manually grouping resources into an AppComponent

Blocked services for manual grouping

AWS Resilience Hub blocks you from manually grouping resources of certain AWS services to prevent configuration errors that could affect the resilience assessment and recommendations for your application. These services are automatically grouped based on their dependencies and configurations. When you define an application inclusive of these resources on AWS Resilience Hub, it analyzes their relationships, dependencies, and resilience requirements to create optimal groupings that ensure accurate assessment results.

List of AWS services blocked for manual grouping:

- Amazon API Gateway
- Amazon DocumentDB
- Amazon DynamoDB
- Amazon Elastic Block Store
- Amazon Elastic File System
- Amazon Relational Database Service
- Amazon S3
- Amazon Simple Queue Service
- FSx for Windows File Server
- NAT Gateway

AWS Resilience Hub resource grouping recommendations

This section explains how to generate and review resource grouping recommendations in AWS Resilience Hub.

> **Note**
>
> You can grant the necessary IAM permissions that are required to work with AWS Resilience Hub by using `AWSResilienceHubAsssessmentExecutionPolicy` AWS managed policy. For more information about AWS managed policy, see AWSResilienceHubAsssessmentExecutionPolicy.

To view resource grouping recommendations

1. In the navigation pane, choose **Applications**.
2. Choose **Add application** page, choose the application name for which you want to review resource grouping recommendations.
3. Choose the **Application structure** tab.
4. If AWS Resilience Hub displays an information alert, choose **Review recommendations** to view all the resource grouping recommendations. Else complete the following steps to manually generate resource grouping recommendations:

   a. Choose **Resources**.

   b. Choose **Get grouping recommendations** from **Actions** menu.

      AWS Resilience Hub scans your resources to check how they can be grouped in the best possible way into relevant AppComponents to improve the accuracy of the assessments. If AWS Resilience Hub learns that your resources can be grouped together, it displays an information alert for the same.

   c. If the information alert is displayed, choose **Review recommendations** to view all the resource grouping recommendations.

   You can identify the AppComponents in the **Review resource grouping recommendations** section using the following:

   - **AppComponent name** – Name of the AppComponent in which the resources will be grouped.

- **Confidence level** – Indicates the confidence level of AWS Resilience Hub in the grouping recommendation.
- **Resource count** – Indicates the number of resources that will be grouped in the AppComponent.
- **AppComponent type** – Indicates the type of AppComponent.

To view resources that will be grouped in AppComponents

1. Complete the steps in **To view resource grouping recommendations** procedure and then return to this procedure.
2. In **Review resource grouping recommendations** section, select the check box (adjacent to the **AppComponent name**) to view all the resources that will be grouped together within the selected AppComponent. If you select multiple check boxes, AWS Resilience Hub displays a dynamically generated **recommendations selected** section that groups the selected AppComponents under their respective AppComponent type. Choose the number below each AppComponent type to view all the resources that will be grouped together within the selected AppComponent.

   You can identify the resources that will be grouped in the selected AppComponent in the **Resources** section using the following:

   - **Logical ID** – Indicates the logical ID of the resource. A logical ID is a name used to identify resources in your AWS CloudFormation stack, Terraform state file, myApplications application, or AWS Resource Groups.
   - **Physical ID** – The actual assigned identifier for the resource, such as an Amazon EC2 instance ID or an Amazon S3 bucket name.
   - **Type** – Indicates the type of resource.
   - **Region** – AWS Region in which the resource is located.

To accept resource grouping recommendations

1. Complete the steps in **To view resource grouping recommendations** procedure and then return to this procedure.
2. In the **Review resource grouping recommendations** section, select all the check boxes adjacent to the **AppComponent name**. To find a specific AppComponent, enter the AppComponent name in the **Find AppComponents** box.

> **Note**
>
> By default, AWS Resilience Hub displays all the resource grouping recommendations. To filter the table with previously rejected resource grouping recommendations, choose **Previously rejected** from the dropdown menu adjacent to the **Find AppComponents** box.

3. Choose **Accept**.
4. Choose **Accept** in the **Accept resource grouping recommendation** dialog.

   AWS Resilience Hub displays an information alert if the resource grouping is successful. If you have accepted only a subset of resource grouping recommendations, **Review resource grouping recommendations** section displays all the resource grouping recommendations that you have not accepted.

To reject resource grouping recommendations

1. Complete the steps in **To view resource grouping recommendations** procedure and then return to this procedure.
2. In **Review resource grouping recommendations** section, select all the check boxes adjacent to the **AppComponent name**. To find a specific AppComponent, enter the AppComponent name in the **Find AppComponents** box.

   > **Note**
   >
   > By default, AWS Resilience Hub displays all the resource grouping recommendations. To filter the table with previously rejected resource grouping recommendations, select **Previously rejected** from the dropdown menu adjacent to the **Find AppComponents** box.

3. Choose **Reject**.
4. Select one of the reasons for rejecting the resource grouping recommendation and then choose **Reject** in the **Reject resource grouping recommendation** dialog.

   AWS Resilience Hub displays an information alert confirming the same. If you have rejected only a subset of resource grouping recommendations, **Review resource grouping**

recommendations section displays all the resource grouping recommendations that you have not accepted.

Manually grouping resources into an AppComponent

This section explains how to manually group resources into an AppComponent and assigning different AppComponent to a resource in AWS Resilience Hub.

To group resources

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the application name that contains the resources that you want to group.
3. Choose the **Application structure** tab.
4. Under **Version** tab, select the application version with **Draft** status.
5. Choose the **Resources** tab.
6. Select the check boxes that is adjacent to **Logical ID** to select all the resources you want to group.

   > **Note**
   >
   > You cannot choose manually added resources.

7. Choose **Actions**, and then choose **Group resources**.
8. Choose an AppComponent from the **Choose AppComponent** dropdown list in which you want to group the resource.
9. Choose **Save**.
10. Choose **Publish new version**.
11. Choose the **Application structure** tab.
12. To view the published version of your application, complete the following steps:

    a. Under **Version** tab, select the application version with **Current release** status.

    b. Choose the **Resources** tab.

To assign resources to an AppComponent

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the application name that contains the resource you want to regroup.
3. Choose the **Application structure** tab.
4. Under **Version**, select the application version with **Draft** status.
5. Choose the **Resources** tab.
6. Select the check box that is adjacent to **Logical ID** to select the resource.
7. Choose **Change AppComponent** from **Actions** menu.
8. To delete the current AppComponent from the **AppComponent** section, choose **X** in the upper-right corner of the label that displays your current AppComponent name.
9. To group the resource in a different AppComponent, choose a different AppComponent from the **Choose AppComponent** dropdown list.
10. Choose **Add**.
11. Delete any empty AppComponents from the **AppComponents** tab.
12. Choose **Publish new version**.
13. Choose the **Application structure** tab.
14. To view the published version of your application, complete the following steps:

    a. Under **Version** tab, select the application version with **Current release** status.

    b. Choose the **Resources** tab.

Publishing a new AWS Resilience Hub application version

After you make changes to your AWS Resilience Hub application resources as described in Editing AWS Resilience Hub application resources, you must publish a new version of your application to run an accurate resiliency assessment. Also, you might need to publish a new version of your application if you added new recommended alarms, SOPs, and tests to your application.

To publish a new version of your application

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the name of the application.
3. Choose the **Application structure** tab.

4. Choose **Publish new version**.
5. In **Publish version** dialog, in the **Name** box, enter a name for the application version or you can use the default name suggested by AWS Resilience Hub.
6. Choose **Publish**.

   When you publish a new version of your application, this becomes the version that is assessed when you run resiliency assessments. Also, the draft version will be identical to the released version until you make any changes.

After you publish a new version of your application, we recommend you to run a new resiliency assessment report to confirm your application still meets your resiliency policy. For information about running an assessment, see Running and managing resiliency assessments in AWS Resilience Hub.

Viewing all the AWS Resilience Hub application versions

To help track the application changes, AWS Resilience Hub displays the previous versions of your application from the time it was created on AWS Resilience Hub.

To view all the versions of your application

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the name of the application.
3. Choose the **Application structure** tab.
4. To view all the previous versions of your application, choose the plus sign (**+**) before **View all versions**. AWS Resilience Hub indicates the draft version and recently released version of your application using **Draft** and **Current release** statuses, respectively. You can choose any version of your application to views its resources, AppComponent, input sources and other associated information.

   In addition, you can also filter the list by using one of the following options:

   - **Filter by version name** – Enter a name to filter the results by the name of your application version.
   - **Filter by a date and time range** – To apply this filter, choose the calendar icon and select one of the following options to filter by the results that matches the time range:
     - **Relative range** – Select one of the available options and choose **Apply**.

If you choose **Custom range** option, enter a duration in **Enter duration** box and select the appropriate unit of time from **Unit of time** dropdown list, then choose **Apply**.

- **Relative range** – To specify the date and time range, provide the start time and end time, and then choose **Apply**.

Viewing resources of AWS Resilience Hub application

To view resources of your application

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, select the application for which you want to update the security permissions.
3. From **Actions**, choose **View resources**.

   In **Resources** tab, you can identify resources in the **Resources** table by the following:

   - **Logical ID** – A logical ID is a name used to identify resources in your AWS CloudFormation stack, Terraform state file, myApplications application, or AWS Resource Groups.

     > **Note**
     >
     > - Terraform lets you use the same name for different resource types. Therefore, you see "- *resource type*" at the end of the logical ID for resources that share the same name.
     > - To view the instances of all the application resources, choose the plus (**+**) sign before the **Logical ID**. To view all the instances of an application resource, choose the plus (**+**) sign before the Logical ID of each resource.
     >
     >   For more information about the supported resources, see the section called "Supported AWS Resilience Hub resources".

   - **Status** – This indicates whether the AWS Resilience Hub will assess your resource for resiliency.
   - **Resource type** – The resource type identifies the component resource for your application. For example, `AWS::EC2::Instance` declares an Amazon EC2 instance. For more

information about grouping AppComponent resources, see Grouping resources in an Application Component.

- **Source name** – The name of the input source. Choose a source name to view its details in the respective application. For manually added input sources, the link will not be available. For example, if you choose the source name that is imported from an AWS CloudFormation stack, you will be redirected to the stack details page on the AWS CloudFormation.
- **Source type** – The type of the input source.
- **AppComponent type** – The type of input source. Input sources include AWS CloudFormation stacks, myApplications applications, AWS Resource Groups, Terraform state files, and manually added resources.

> **Note**
>
> To edit your Amazon EKS clusters, complete the steps in **To edit the input sources of your AWS Resilience Hub application** procedure.

- **Physical ID** – The actual assigned identifier for that resource, such as an Amazon EC2 instance ID or an S3 bucket name.
- **Included** – This indicates whether AWS Resilience Hub includes these resources in the application.
- **AppComponents** – The AWS Resilience Hub component that was assigned to this resource when its application structure was discovered.
- **Name** – Name of the application resource.
- **Account** – The AWS account that owns the physical resource.

4. Choose **Save and update**.

Deleting an AWS Resilience Hub application

After you've reached the maximum limit of 50 applications, you must delete one or more applications before you can add more.

To delete an application

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, select the application that you want to delete.
3. Choose **Actions**, and then choose **Delete application**.

4. To confirm the deletion, enter **Delete** in the **Delete** box, and choose **Delete**.

Application configuration parameters

AWS Resilience Hub provides an input mechanism to gather additional information about the resources associated with your applications. With this information, AWS Resilience Hub will gain a deeper understanding of your resources and provide better resiliency recommendations.

The **Application configuration parameters** section lists all the configuration parameters of your cross-Region failover support for AWS Elastic Disaster Recovery. You can identify the configuration parameters by the following:

- **Topic** – Indicates the area of your application that is configured. For example, failover configuration.
- **Purpose** – Indicates the reason why AWS Resilience Hub requested the information.
- **Parameter** – Indicates the details that are specific to the area of application, which AWS Resilience Hub will be using to provide recommendations for your application. Currently, this parameter uses a key-value of only one failover Region and one associated account.

Updating application configuration parameters

This section allows you to update the configuration parameters of your AWS Elastic Disaster Recovery and publish the application to include the updated parameters for resiliency assessments.

To update application configuration parameters

1. In the navigation pane, choose **Applications**.
2. On the **Applications** page, choose the application name that you want to edit.
3. Choose the **Application configuration parameters** tab.
4. Choose **Update**.
5. Enter the failover account ID in the **Account ID** box.
6. Select a failover Region from the **Region** dropdown list.

> **Note**
>
> If you want to disable this feature, select "**–**" from the dropdown list.

7. Choose **Update and publish**.

Managing resiliency policies

This section describes how to create resiliency policies for your applications. Setting resiliency policies correctly enables you to understand your application's resiliency posture. A resiliency policy contains information and objectives that you use to assess whether your application is estimated to recover from a disruption type, such as software, hardware, Availability Zone, or AWS Region. These policies do not change or affect an actual application. Multiple applications can have the same resiliency policy.

When you create a resiliency policy, you define the target objectives: recovery time objective (RTO) and recovery point objective (RPO). The objectives determine whether the application meets the resiliency policy. Attach the policy to your application and run a resiliency assessment. You can create different policies for the different types of applications in your portfolio. For example, a real-time trading application would have a different resiliency policy than a monthly reporting application.

> **Note**
>
> AWS Resilience Hub allows you to enter a value zero in the **RTO** and **RPO** fields of your resiliency policy. But, while assessing your application, the lowest possible assessment result is near zero. Hence, if you enter a value zero in the **RTO** and **RPO** fields, the estimated workload RTO and estimated workload RPO result will be near zero and the **Compliance status** for your application will be set to **Policy breached**.

The assessment evaluates your application configuration against the attached resiliency policy. At the end of the process, AWS Resilience Hub provides an assessment of how your application measures against the recovery targets in your resiliency policy.

You can create resiliency policies in Applications, and also in Resiliency policies. You can access relevant details about your policies, and also modify and delete them.

AWS Resilience Hub uses your RTO and RPO targets to measure resiliency for these potential types of disruptions:

- **Application** – Loss of a required software service or process.

- **Cloud infrastructure** – Loss of hardware, such as EC2 instances.
- **Cloud infrastructure Availability Zone (AZ)** – One or more Availability Zones are unavailable.
- **Cloud infrastructure Region** – One or more Regions are unavailable.

AWS Resilience Hub enables you to create customized resiliency policies or use our recommended, open standard resiliency policies. When you create customized policies, name and describe your policy and choose the appropriate level or tier that defines your policy. These tiers include: Foundational IT core services, Mission critical, Critical, Important, and Non-critical.

Choose the tier that is appropriate for your class of application. For example, you might classify a real-time trading system as critical, while you might classify a monthly reporting application as non-critical. When you use our standard policies, you can choose a resiliency policy with a preconfigured tier and values for the RTO and RPO targets by disruption type. If necessary, you can change the tier and the RTO and RPO targets.

You can create resiliency policies in Resiliency policies, or when you describe a new application.

Creating resiliency policies

In AWS Resilience Hub, you can create a resiliency policy. A resiliency policy contains information and objectives that you use to assess whether your application can recover from a disruption type, such as software, hardware, Availability Zone, or AWS Region. These policies do not change or affect an actual application. Multiple applications can have the same resiliency policy.

When you create a resiliency policy, you define the recovery time objective (RTO) and recovery point objective (RPO) targets. When you run an assessment, AWS Resilience Hub determines whether the application is estimated to meet the objectives that are defined in the resiliency policy.

The assessment evaluates your application configuration against the attached resiliency policy. At the end of the process, AWS Resilience Hub provides an assessment of how your application measures against the objectives in your resiliency policy.

> **Note**
>
> AWS Resilience Hub allows you to enter a value zero in the **RTO** and **RPO** fields of your resiliency policy. But, while assessing your application, the lowest possible assessment result is near zero. Hence, if you enter a value zero in the **RTO** and **RPO** fields, the

estimated workload RTO and estimated workload RPO result will be near zero and the **Compliance status** for your application will be set to **Policy breached**.

You can create resiliency policies in Applications, and also in Resiliency policies. You can access relevant details about your policies, and also modify and delete them.

To create resiliency policies in Applications

1. In the left navigation menu, choose **Applications**.
2. Complete the procedures from the section called "Get started by adding an application" through the section called "Add tags to your application ".
3. In **Resiliency policies** section, choose **Create resiliency policy**.

   The **Create resiliency policy** page displays.
4. In the **Choose a creation method** section, select **Create a policy**.
5. Enter a name for the policy.
6. (Optional) Enter a description for the policy.
7. Choose one of the following from **Tier** dropdown list:

   - **Foundational IT core services**
   - **Mission critical**
   - **Critical**
   - **Important**
   - **Non critical**
8. For both **RTO** and **RPO** targets, under **Customer Application RTO and RPO**, enter a numeric value in the box, and then choose the unit of time that the value represents.

   Repeat these entries under **Infrastructure RTO and RPO** for **Infrastructure** and **Availability Zone**.
9. (Optional) If you have a multi-Region application, you may want to define a Region's RTO and RPO targets.

   Turn-on **Region**. For both Region **RTO** and **RPO** targets, under **Customer Application RTO and RPO**, enter a numeric value in the box, and then choose the unit of time that the value represents.

10. (Optional) If you want to add tags, you can do that later as you continue creating your policy. For more information about tags, see Tagging resources in the *AWS General Reference*.
11. To create the policy, choose **Create**.

To create resiliency policies in Resiliency policies

1. In the left navigation menu, choose **Policies**.
2. In **Resiliency policies** section, choose **Create resiliency policy**.

   The **Create resiliency policy** page displays.
3. Enter a name for the policy.
4. (Optional) Enter a description for the policy.
5. Choose one of the following from **Tier**:

   - **Foundational IT core services**
   - **Mission critical**
   - **Critical**
   - **Important**
   - **Non critical**
6. For both **RTO** and **RPO** targets, under **Customer Application RTO and RPO**, enter a numeric value in the box and then choose the unit of time that the value represents.

   Repeat these entries under **Infrastructure RTO and RPO** for **Infrastructure** and **Availability Zone**.
7. (Optional) If you have a multi-Region application, you may want to define a Region's RTO and RPO targets.

   Turn-on **Region**. For both **RTO** and **RPO** targets, under **Customer Application RTO and RPO**, enter a numeric value in the box and then choose the unit of time that the value represents.
8. (Optional) If you want to add tags, you can do that later as you continue creating your policy. For more information about tags, see Tagging resources in the *AWS General Reference*.
9. To create the policy, choose **Create**.

To create resiliency policies based on a suggested policy

1. In the left navigation menu, choose **Policies**.
2. In the **Choose a creation method** section, select **Select a policy based on a suggested policy**.
3. In **Resiliency policies** section, choose **Create resiliency policy**.

   The **Create resiliency policy** page displays.
4. Enter a name for the resiliency policy.
5. (Optional) Enter a description for the policy.
6. Under **Suggested resiliency policies** section, view and choose one of the following predetermined resiliency policy tiers:

   - **Non-critical application**
   - **Important Application**
   - **Critical Application**
   - **Global Critical Application**
   - **Mission Critical Application**
   - **Global Mission Critical Application**
   - **Foundational Core Service**
7. To create the resiliency policy, choose **Create policy**.

Accessing resiliency policy details

When you open a resiliency policy, you see important details about the policy. You can also edit or delete the resiliency.

Resiliency policy details consist of two major views: **Summary** and **Tags**.

Summary

Basic information

Provides the following information about resiliency policy: Name, Description, Tier, Cost Tier, and Date Created.

Estimated workload RTO and estimated workload RPO

Shows the estimated workload RTO and estimated workload RPO disruption type associated with this resiliency policy.

Tags

Use this view to manage, add, and delete tags internal to this application.

To edit resiliency policies in Resiliency policy details

1. In the left navigation menu, choose **Policies**.
2. In **Resiliency policies**, open a resiliency policy.
3. Choose **Edit**. Enter appropriate changes in **Basic Info**, and **RTO** and **RPO** fields. Then choose **Save changes**.

To edit resiliency policies in Resiliency policy

1. In the left navigation menu, choose **Policies**.
2. In **Resiliency policies**, choose a resiliency policy.
3. Choose **Actions**, and then select **Edit**.
4. Enter appropriate changes in **Basic Info**, and **RTO** and **RPO** fields. Then choose **Save changes**.

To delete resiliency policies in Resiliency policy details

1. In the left navigation menu, choose **Policies**.
2. In **Resiliency policies**, open a resiliency policy.
3. Choose **Delete**. Confirm your deletion, and then choose **Delete**.

To delete resiliency policies in Resiliency policy

1. In the left navigation menu, choose **Policies**.
2. In **Resiliency policies**, choose a resiliency policy.
3. Choose **Actions**, and then select **Delete**.
4. Confirm your deletion, and then choose **Delete**.

Running and managing resiliency assessments in AWS Resilience Hub

When your application changes, you should run a resiliency assessment. The assessment compares each Application Component configuration to the policy and makes alarm, SOP, and test recommendations. These configuration recommendations can improve the speed of recovery procedures.

Alarm recommendations help you set alarms that detect outages. SOP recommendations provide scripts that manage common recovery processes, such as recovery from a backup. Test recommendations offer suggestions to verify your configurations work properly. For example, you can test whether an application recovers during automatic recovery processes, such as automatic scaling or load balancing because of network issues. You can test whether application alarms are triggered when resources reach their limits. You can also test how well SOPs work under the conditions that you indicate.

Topics:

- Running resiliency assessments in AWS Resilience Hub
- Reviewing assessments reports
- Deleting resiliency assessments

Running resiliency assessments in AWS Resilience Hub

You can run resiliency assessments from multiple locations in AWS Resilience Hub. For more information about your application, see the section called “Managing applications”.

To run a resiliency assessment from the Actions menu

1. In the left navigation menu, choose **Applications**.
2. Choose an application from the **Applications** table.
3. Choose the **Assess resiliency** from the **Actions** menu.
4. In **Run resiliency assessment** dialog, you can enter a unique name or use the generated name for the assessment.
5. Choose **Run**.

   To review the assessment report, choose **Assessments** in your application. For more information, see the section called “Reviewing assessments reports”.

To run a resiliency assessment from the Assessments tab

You can run a new resiliency assessment when your application or resiliency policy changes.

1. In the left navigation menu, choose **Applications**.
2. Choose an application from the **Applications** table.
3. Choose the **Assessments** tab.
4. Choose **Run resiliency assessment**.
5. In **Run resiliency assessment** dialog, you can enter a unique name or use the generated name for the assessment.
6. Choose **Run**.

   To review the assessment report, choose **Assessments** in your application. For more information, see the section called “Reviewing assessments reports”.

Reviewing assessments reports

You find assessment reports in the **Assessments** view of your application.

To find an assessment report

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application.
3. In **Assessments** tab, choose an assessment report from the **Resiliency assessments** section.

When you open the report, you see the following:

- An overall overview of the assessment report
- Recommendations to improve resiliency.
- Recommendations to set up alarms, SOPs, and tests
- How to create and manage tags to search and filter your AWS resources

Assessment report

This section provides an overview of the assessment report. AWS Resilience Hub lists each disruption type and the associated Application Component. It also lists your actual RTO and RPO policies and determines whether the Application Component can achieve the policy goals.

Overview

Shows the name of the application, the name of the resiliency policy, and the creation date of the report.

Detected resource drifts

This section lists all the resources that were added or removed after they were included in the latest version of the published application. Choose **Reimport input sources** to reimport all the input sources (which contains drifted resources) in the **Input sources** tab. Choose **Publish and assess** to include the updated resources in the application and receive an accurate resiliency assessment.

You can identify the drifted input sources using the following:

- **Logical ID** – Indicates the logical ID of the resource. A logical ID is a name used to identify resources in your AWS CloudFormation stack, Terraform state file, myApplications application, or AWS Resource Groups.
- **Change** – Indicates if an input resource was **Added** or **Removed**.
- **Source name** – Indicates the resource name. Choose a source name to view its details in the respective application. For manually added input sources, the link will not be available. For example, if you choose the source name that is imported from an AWS CloudFormation stack, you will be redirected to the stack details page on the AWS CloudFormation.
- **Resource type** – Indicates the resource type.
- **Account** – Indicates the AWS account that owns the physical resource.
- **Region** – Indicates the AWS Region where the resource is located.

RTO

Shows a graphical representation of whether the application is estimated to meet resiliency policy's objectives. This is based on the amount of time that an application can be down without causing significant damage to the organization. The assessment provides an estimated workload RTO.

RPO

Shows a graphical representation of whether the application is estimated to meet resiliency policy's objectives. This is based on the amount of time that data can be lost before a significant harm to the business occurs. The assessment provides an estimated workload RPO.

Details

Provides detailed descriptions of each disruption type using **All results** and **Application compliance drifts** tabs. **All results** tab shows all the disruptions including compliance drifts, and **Application compliance drifts** tab displays only compliance drifts. Disruption type includes **Application**, cloud infrastructure (**Infrastructure** and **Availability Zone**), and **Region**, and provides the following information about it:

- **AppComponent**

  The resources that comprise the application. For example, your application might have a database or compute component.

- **Estimated RTO**

  Indicates whether your policy configuration aligns with your policy requirement. We provide two values, our **Estimated RTO** and your **Targeted RTO**. For example, if you see **2h** value under **Targeted RTO** and **40m** under **Estimated Workload RTO**, it indicates that we provide an estimated workload RTO of 40 minutes, while the current RTO of your application is two hours. We base our estimated workload RTO calculation on the configuration, not the policy. As a result, a multi-Availability Zone database will have the same estimated workload RTO for Availability Zone failure, no matter which policy you select.

- **RTO drift**

  Indicates the duration by which your application has drifted from the estimated workload RTO of the previous successful assessment. We provide two values, our **Estimated RTO** and **RTO drift**. For example, if you see **2h** value under **Estimated RTO** and **40m** under **RTO drift**, it indicates that your application drifts from the estimated workload RTO of the previous successful assessment by 40 minutes.

- **Estimated RPO**

Shows the actual **Estimated Workload RPO** policy that AWS Resilience Hub estimates, based on the **Targeted RPO** policy that you set for each Application Component. For example, you might have set the RPO target in your resiliency policy for Availability Zone failures to one hour. The estimated result might be calculated near to zero. This assumes that Amazon Aurora, where we commit every transaction, is successful in four out of six nodes, spanning multiple Availability Zones. It might be five minutes for point-in-time restore.

The only RTO and RPO target that you can opt not to supply is Region. For some applications, it is useful to plan for recovery when there is a crucial dependency on an AWS service, which might become unavailable in the entire Region.

If you choose this option, such as setting RTO or RPO targets for the Region, you'll receive an estimated recovery time and operational recommendations for such failures.

- **RPO drift**

  Indicates the duration by which your application has drifted from the estimated workload RPO of the previous successful assessment. We provide two values, our **Estimated RPO** and **RPO drift**. For example, if you see **2h** value under **Estimated RPO** and **40m** under **RPO drift**, it indicates that your application drifts from the estimated workload RPO of the previous successful assessment by 40 minutes.

Reviewing resiliency recommendations

Resiliency recommendations evaluate Application Components and recommend how to optimize by estimated workload RTO and estimated workload RPO, costs, and minimal changes.

With AWS Resilience Hub, you can optimize resiliency using one of the following recommended options in **Why you should choose this option**:

> **Note**
>
> - AWS Resilience Hub provides up to three AWS Resilience Hub recommended options.
> - If you set Regional RTO and RPO targets, AWS Resilience Hub displays **Optimize for Region RTO/RPO** in the recommended options. If Regional RTO and RPO targets are not set, **Optimize for Availability Zone (AZ) RTO/RPO** is displayed. For more information about setting Regional RTO/RPO targets while creating resiliency policies, see Creating resiliency policies.

- Estimated workload RTO and estimated workload RPO values for the applications and their configurations are determined by considering the amount of data and individual AppComponents. However, these values are only estimates. You should use your own testing (such as AWS Fault Injection Service) to test your application for actual recovery times.

Optimize for Availability Zone RTO/RPO

The lowest possible estimated workload recovery times (RTO/RPO) during an Availability Zone (AZ) disruption. If your configuration can't be changed sufficiently to meet the RTO and RPO targets, you're informed about the lowest estimated workload AZ recovery times to get your configuration close to the possibility of meeting the policy.

Optimize for Region RTO/RPO

The lowest possible estimated workload recovery times (RTO/RPO) during a Regional disruption. If your configuration can't be changed sufficiently to meet the RTO and RPO targets, you're informed about the lowest estimated workload Region recovery times to get your configuration close to the possibility of meeting the policy.

Optimize for cost

The lowest cost that you can incur and still meet your resiliency policy. If your configuration can't be changed sufficiently to meet the optimization goals, you're informed about the lowest cost that you can incur to get your configuration close to the possibility of meeting the policy.

Optimize for minimal changes

The minimum changes required to achieve your policy targets. If your configuration can't be changed sufficiently to meet the optimization goals, you're informed about the recommended changes that can get your configuration close to the possibility of meeting the policy.

The following items are included in the optimization category breakdowns:

- **Description**

  Describes the configurations suggested by AWS Resilience Hub.
- **Changes**

A list of text changes that describe the necessary tasks to switch to the suggested configuration.

- **Base cost**

  The estimated cost associated with the recommended changes.

  > **Note**
  >
  > **Base cost** can vary based on the usage and it does not include any discounts or offers from Enterprise Discount Program (EDP).

- **Estimated Workload RTO and RPO**

  The estimated workload RTO and estimated workload RPO after changes.

AWS Resilience Hub evaluates whether an Application Component (AppComponent) can comply with a resiliency policy. If the AppComponent does not comply with a resiliency policy and AWS Resilience Hub cannot make any recommendations to facilitate compliance, it might be because the recovery time for the selected AppComponent cannot be met within the constraints of the AppComponent. Examples of AppComponent constraints include resource type, storage size, or resource configuration.

To facilitate the compliance of the AppComponent with the resiliency policy, change the resource type of the AppComponent or update the resiliency policy to align with what the resource can deliver.

Reviewing operational recommendations

Operational recommendations contain recommendations to set up alarms, SOPs, and AWS FIS experiments through AWS CloudFormation templates.

AWS Resilience Hub provides AWS CloudFormation template files for you to download and manage the application's infrastructure as code. As a result, we supply recommendations in AWS CloudFormation so that you can add them to your application code. If the size of AWS CloudFormation template file is more than one MB and contains more than 500 resources, AWS Resilience Hub generates more than one AWS CloudFormation template file where the size of each file is not more than one MB and contains up to 500 resources. If the AWS CloudFormation template file is split into multiple files, the AWS CloudFormation template file names will be appended with `partXofY`, where X denotes the file number in the sequence and Y denotes the

total number of files the AWS CloudFormation template file is divided into. For example, if the template file `big-app-template5-Alarm-104849185070-us-west-2.yaml` is divided into four files, the file names would be as follows:

- `big-app-template5-Alarm-104849185070-us-west-2-part1of4.yaml`
- `big-app-template5-Alarm-104849185070-us-west-2-part2of4.yaml`
- `big-app-template5-Alarm-104849185070-us-west-2-part3of4.yaml`
- `big-app-template5-Alarm-104849185070-us-west-2-part4of4.yaml`

However, in case of large AWS CloudFormation templates, you are requested to provide the Amazon Simple Storage Service URI instead of using CLI/API with local file as input.

In AWS Resilience Hub, you can perform the following actions:

- You can provision the selected alarms, SOPs, and AWS FIS experiments. To provision alarms, SOPs, and AWS FIS experiments, select the appropriate recommendation and enter a unique name. AWS Resilience Hub creates a template based on your selected recommendations. In **Templates**, you can access your created templates through an Amazon Simple Storage Service (Amazon S3) URL.
- You can include or exclude selected alarms, SOPs, and AWS FIS experiments that were recommended for your application at any point of time. For more information see, the section called “Including or excluding operational recommendations”.
- You can also search, create, add, remove, and manage tags, for an application and see all the tags associated with it.

Including or excluding operational recommendations

AWS Resilience Hub provides an option to include or exclude the alarms, SOPs, and AWS FIS experiments (tests) that were recommended for improving the resiliency score of your application at any point of time. Including and excluding operational recommendations will have an impact on the resiliency score of your application only after you run a new assessment. Hence, we recommend you to run an assessment to get the updated resiliency score and understand its impact on your application.

For more information about restricting permissions to include or exclude recommendations per application, see the section called “Limiting permissions to include or exclude AWS Resilience Hub recommendations”.

To include or exclude operational recommendations from applications

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application.
3. Choose **Assessments** and select an assessment from the **Resiliency assessments** table. If you don't have an assessment, complete the procedure in the section called “Running resiliency assessments in AWS Resilience Hub” and then return to this step.
4. Select **Operational recommendations** tab.
5. To include or exclude operational recommendations from your application, complete the following procedures:

To include or exclude recommended alarms from your application

1. To exclude alarms, complete the following steps:

   a. Under **Alarms** tab, from **Alarms** table, select all the alarms (with **Not implemented** state) you want to exclude. You can identify the current implementation state of an alarm from the **State** column.

   b. From **Actions**, choose **Exclude selected**.

   c. From **Exclude recommendations** dialog, select one of the following reasons (optional), and choose **Exclude selected** to exclude the selected alarms from the application.

      - **Already implemented** – Choose this option if you have already implemented these alarms in an AWS service such as Amazon CloudWatch, or any other third-party service provider.
      - **Not relevant** – Choose this option if the alarms do not suit your business requirements.
      - **Too complicated to implement** – Choose this option if you think these alarms are too complicated to implement.
      - **Other** – Choose this option to specify any other reason for excluding the recommendation.

2. To include alarms, complete the following steps:

   a. Under **Alarms** tab, from **Alarms** table, select all the alarms (with **Excluded** state) you want to include. You can identify the current implementation state of the alarm from the **State** column.

   b. From **Actions**, choose **Include selected**.

c. From **Include recommendations** dialog, choose **Include selected** to include all the selected alarms in your application.

To include or exclude recommended standard operating procedures (SOPs) from your application

1. To exclude recommended SOPs, complete the following steps:

   a. Under **Standard operating procedures** tab, from **SOPs** table, select all the SOPs (with **Implemented** or **Not implemented** state) you want to exclude. You can identify the current implementation state of an SOP from the **State** column.

   b. From **Actions**, choose **Exclude selected** to exclude the selected SOPs from your application.

   c. From **Exclude recommendations** dialog, select one of the following reasons (optional), and choose **Exclude selected** to exclude the selected SOPs from the application.

      - **Already implemented** – Choose this option if you have already implemented these SOPs in an AWS service, or any other third-party service provider.
      - **Not relevant** – Choose this option if the SOPs do not suit your business requirements.
      - **Too complicated to implement** – Choose this option if you think these SOPs are too complicated to implement.
      - **None** – Choose this option if you don't want to specify the reason.

2. To include SOPs, complete the following steps:

   a. Under **Standard operating procedures** tab, from **SOPs** table, select all the alarms (with **Excluded** state) you want to include. You can identify the current implementation state of the alarm from the **State** column.

   b. From **Actions**, choose **Include selected**.

   c. From **Include recommendations** dialog, choose **Include selected** to include all the selected SOPs in your application.

To include or exclude recommended tests from your application

1. To exclude recommended tests, complete the following steps:

a. Under **Fault injection experiment templates** tab, from **Fault injection experiment templates** table, select all the tests (with **Implemented** or **Not implemented** state) you want to exclude. You can identify the current implementation state of a test from the **State** column.

b. From **Actions**, choose **Exclude selected**.

c. From **Exclude recommendations** dialog, select one of the following reasons (optional), and choose **Exclude selected** to exclude the selected AWS FIS experiments from the application.

- **Already implemented** – Choose this option if you have already implemented these tests in an AWS service, or any other third-party service provider.
- **Not relevant** – Choose this option if the tests do not suit your business requirements.
- **Too complicated to implement** – Choose this option if you think these tests are too complicated to implement.
- **None** – Choose this option if you don't want to specify the reason.

2. To include recommended tests, complete the following steps:

a. Under **Fault injection experiment templates** tab, from **Fault injection experiment templates** table, select all the tests (with **Excluded** state) you want to include. You can identify the current implementation state of the test from the **State** column.

b. From **Actions**, choose **Include selected**.

c. From **Include recommendations** dialog, choose **Include selected** to include all the selected tests in your application.

Deleting resiliency assessments

You can delete resiliency assessments in the **Assessments** view of your application.

To delete a resiliency assessment

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application.
3. In **Assessments**, choose an assessment report in the **Resiliency assessments** table.
4. To confirm the deletion, choose **Delete**.

The report no longer appears in the **Resiliency assessments** table.

Running and managing resiliency assessments from Resiliency widget

AWS Resilience Hub enables you to run assessments for applications created and managed in myApplications in Resiliency widget. Whenever you make modifications to an application, it is recommended to run a resiliency assessment from Resiliency widget or from AWS Resilience Hub console. During this assessment, the configuration of each Application Component is evaluated against established policies and best practices. Based on this evaluation, the assessment generates recommendations for setting up alarms, creating Standard Operating Procedures (SOPs), and implementing testing strategies. Implementing these configuration recommendations can enhance the speed and efficiency of your recovery procedures, ensuring faster incident response and minimizing potential downtime.

Alarm recommendations help you set alarms that detect outages. SOP recommendations provide scripts that manage common recovery processes, such as recovery from a backup. Test recommendations offer suggestions to verify your configurations work properly. For example, you can test whether an application recovers during automatic recovery processes, such as automatic scaling or load balancing because of network issues. You can test whether application alarms are triggered when resources reach their limits. You can also test how well SOPs work under the conditions that you indicate.

Topics:

- Running resiliency assessments from Resiliency widget
- Reviewing assessment summary in Resiliency widget

Running resiliency assessments from Resiliency widget

For applications created in **myApplications** widget, you can now run resiliency assessments from the **Resiliency** widget and AWS Resilience Hub console. For more information about running resiliency assessments from AWS Resilience Hub console, see Running resiliency assessments in AWS Resilience Hub.

To run a resiliency assessment for an existing myApplications application from Resiliency widget for the first time

1. Sign in to the AWS Management Console.

2. Expand the left sidebar and choose **myApplications**.
3. Select the application for which you want to run assessment.

   As a prerequisite, ensure that you have added the **Resiliency** widget in your AWS Console. To add this widget, complete the following steps.

   a. On the upper or lower right of the **Console Home** dashboard, choose **+Add widgets**.

   b. Choose the **drag indicator**, represented by six vertical dots in the upper left of the widget title bar, and then drag it to your **Console Home** dashboard.
4. Choose **Assess application**.
5. To select an existing IAM role that will be used for accessing resources in the current account, select **Use an IAM role** and then select an IAM role from the **Select an IAM role** dropdown list.

   If you want to use current IAM user to discover your application resources, choose **Use the current IAM user permissions** and select **I understand that I must manually configure permissions to enable the required functionality within AWS Resilience Hub** in **Use the current IAM user to discover application resources** section.
6. Choose **Assess**.

   Alternatively, turn on **Automatically assess daily** to enable AWS Resilience Hub to assess your application daily without any additional costs.

   AWS Resilience Hub performs the following actions:

   - Creates an application in AWS Resilience Hub and automatically discovers and maps the associated resources.
   - Creates and assigns a new resiliency policy with pre-defined values for recovery time objective (RTO) and recovery point objective (RPO). That is, four hours for RTO and one hour for RPO. After you generate an assessment, you can modify the resiliency policy or assign a different policy from the AWS Resilience Hub console. For more information about updating resiliency policy and attaching a different policy, see Managing resiliency policies.
   - Assesses the resilience of the application against RTO and RPO, and continuously monitors resources and configuration changes, and publishes the results.

> **Note**
>
> Before starting assessments, it is advisable to evaluate the potential costs involved in running assessments using AWS Resilience Hub. For detailed pricing information, see the AWS Resilience Hub pricing.

To rerun a resiliency assessment for an existing myApplications application from Resiliency widget

1. Sign in to the AWS Management Console.
2. Expand the left sidebar and choose **myApplications**.
3. Select the application you want to reassess.

   As a prerequisite, ensure that you have added the **Resiliency** widget in your AWS Console. To add this widget, complete the following steps.

   a. On the upper or lower right of the **Console Home** dashboard, choose **+Add widgets**.

   b. Choose the **drag indicator**, represented by six vertical dots in the upper left of the widget title bar, and then drag it to your **Console Home** dashboard.
4. Choose **Reassess** from the **Resiliency** widget.

   Alternatively, turn on **Automatically assess daily** to enable AWS Resilience Hub to assess your application daily without any additional costs.

Reviewing assessment summary in Resiliency widget

The **Resiliency** widget displays a snapshot of the assessment results that will provide you with most important and actionable insights into the myApplications application's resilience, potential vulnerabilities, key performance indicators (KPIs) and recommended actions for improvement. You can learn more about the application's resiliency posture from the most recent assessment using the following:

- **Resiliency score history** – This chart displays the trend of the application's resiliency score for up to one year.

- **Resiliency score** – Indicates the resiliency score of the application evaluated in the latest assessment. This score reflects how closely your application follows our recommendations for meeting the application's resiliency policy, and for implementing alarms, standard operating procedures (SOPs), and AWS Fault Injection Service (AWS FIS) experiments. Choose the number to view additional information in the **Resiliency score** section under **Summary** tab in the AWS Resilience Hub console. For more information, see Assessment report.
- **Policy breaches** – Choose the number below to view all the Application Components (AppComponents) that breaches the policies attached to your application in the **Assessment report** pane in the AWS Resilience Hub console. For more information, see Assessment report.
- **Policy drifts** – Indicates the AppComponents that complied with the policy in the previous assessment but failed to comply in the current assessment. Choose the number below to view the AppComponents in the **Assessment report** pane in the AWS Resilience Hub console. For more information, see Assessment report.
- **Resource drifts** – Choose the number below to view all the resources that drifted from the latest assessment in the **Assessment report** pane in the AWS Resilience Hub console. For more information, see Assessment report.
- **Go to Resilience Hub** – Choose this option to open your application in the AWS Resilience Hub console.

Managing alarms

When you run a resiliency assessment, as a part of operational recommendations, AWS Resilience Hub recommends setting up Amazon CloudWatch alarms to monitor your application resiliency. We recommend these alarms based on the resources and components of your current application configuration. If the resources and components in your application change, you should run a resiliency assessment to ensure you have the correct Amazon CloudWatch alarms for your updated application.

Additionally, AWS Resilience Hub now automatically detects and integrates any already configured Amazon CloudWatch alarms into its resilience assessments, providing a more comprehensive view of your application's resilience posture. This new capability combines AWS Resilience Hub recommendations with your current monitoring setup, streamlining alarm management and enhancing assessment accuracy. If you have implemented an Amazon CloudWatch alarm and AWS Resilience Hub doesn't automatically detect it, you can exclude the alarm and select the reason as **Already implemented**. For more information about excluding recommendation, see Including or excluding operational recommendations.

AWS Resilience Hub provides a template file (`README.md`) that allows you to create alarms recommended by AWS Resilience Hub within AWS (such as Amazon CloudWatch) or outside AWS. The default values provided in the alarms are based on the best practices that are used for creating these alarms.

Topics

- Creating alarms from the operational recommendations
- Viewing alarms

Creating alarms from the operational recommendations

AWS Resilience Hub creates an CloudFormation template that contains details to create the selected alarms in Amazon CloudWatch. After the template is generated, you can access it through an Amazon S3 URL, download the same and place it in your code pipeline or create a stack through the CloudFormation console.

To create an alarm based on AWS Resilience Hub recommendations, you must create an CloudFormation template for the recommended alarms and include them in your code base.

To create alarms in operational recommendations

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, choose your application.
3. Choose **Assessments** tab.

   In **Resiliency assessments** table, you can identify your assessments using the following information:

   - **Name** – Name of the assessment you had provided at the time of creation.
   - **Status** – Indicates the execution state of the assessment.
   - **Compliance status** – Indicates if the assessment is compliant with the resiliency policy.
   - **Resiliency drift status** – Indicates if your application has drifted or not from the previous successful assessment.
   - **App version** – Version of your application.
   - **Invoker** – Indicates the role that invoked the assessment.
   - **Start time** – Indicates the start time of the assessment.

- **End time** – Indicates the end time of the assessment.
- **ARN** – The Amazon Resource Name (ARN) of the assessment.

4. Select an assessment from the **Resiliency assessments** table. If you don't have an assessment, complete the procedure in the section called "Running resiliency assessments in AWS Resilience Hub" and then return to this step.
5. Choose **Operational recommendations**.
6. If not selected by default, choose **Alarms** tab.

   In **Alarms** table, you can identify the recommended alarms using the following:

   - **Name** – Name of the alarm that you have set for your application.
   - **Description** – Describes the objective of the alarm.
   - **State** – Indicates the current implementation state of the Amazon CloudWatch alarms.

     This column displays one of the following values:

     - **Implemented** – Indicates that the alarms recommended by AWS Resilience Hub are implemented in your application. Choosing the number below will filter the **Alarms** table to display all the recommended alarms that are implemented in your application.
     - **Not implemented** – Indicates that the alarms recommended by AWS Resilience Hub are included but not implemented in your application. Choosing the number below will filter the **Alarms** table to display all the recommended alarms that are not implemented in your application.
     - **Excluded** – Indicates that the alarms recommended by AWS Resilience Hub are excluded from your application. Choosing the number below will filter the **Alarms** table to display all the recommended alarms that are excluded from your application. For more information about including and excluding recommended alarms, see Including or excluding operational recommendations.
     - **Inactive** – Indicates that the alarms are deployed to Amazon CloudWatch, but the status is set to **INSUFFICIENT_DATA** in Amazon CloudWatch. Choosing the number below will filter the **Alarms** table to display all the implemented and inactive alarms.
   - **Configuration** – Indicates if there are any pending configuration dependencies that needs to be addressed.
   - **Type** – Indicates the type of alarm.
   - **AppComponent** – Indicates the Application Components (AppComponents) that are associated with this alarm.

- **Reference ID** – Indicates the logical identifier of the AWS CloudFormation stack event in AWS CloudFormation.
- **Recommendation ID** – Indicates the logical identifier of the AWS CloudFormation stack resource in AWS CloudFormation.

7. In **Alarms** tab, to filter the alarm recommendations in **Alarms** table based on a specific state, select a number below the same.
8. Select the recommended alarms that you want to set up for your application, and choose **Create CloudFormation template**.
9. In **Create CloudFormation template** dialog, you can use the auto-generated name, or you can enter a name for CloudFormation template in the **CloudFormation template name** box.
10. Choose **Create**. This can take up to a few minutes to create the AWS CloudFormation template.

Complete the following procedure to include the recommendations in your code base.

To include the AWS Resilience Hub recommendations your code base

1. Choose **Templates** tab to view the template you just created. You can identify your templates using the following:
   - **Name** – Name of the assessment you had provided at the time of creation.
   - **Status** – Indicates the execution state of the assessment.
   - **Type** – Indicates the type of operational recommendation.
   - **Format** – Indicates the format (JSON/ text) in which the template is created.
   - **Start time** – Indicates the start time of the assessment.
   - **End time** – Indicates the end time of the assessment.
   - **ARN** – The ARN of the template
2. Under **Template details**, choose the link below **Templates S3 Path** to open the template object in Amazon S3 console.
3. In Amazon S3 console, from **Objects** table, choose the Alarms folder link.
4. To copy the Amazon S3 path, select the check box in front of the JSON file and choose **Copy URL**.
5. Create an AWS CloudFormation stack from AWS CloudFormation console. For more information about creating an AWS CloudFormation stack, see https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-console-create-stack.html.

While creating the AWS CloudFormation stack, you must provide the Amazon S3 path that you copied from the previous step.

Viewing alarms

You can view all the active alarms that you have set up to monitor the resiliency of your applications. AWS Resilience Hub uses CloudFormation template to store alarm details that is in-turn used for creating the alarms in Amazon CloudWatch. You can access the CloudFormation template using Amazon S3 URL, and can download and place it into your code pipeline or create a stack through the CloudFormation console.

To view alarms from the dashboard, choose **Dashboard** from the left navigation menu. In **Implemented alarms** table, you can identify the implemented alarms using the following information:

- **Application impacted** – Name of the applications that have implemented this alarm.
- **Active alarms** – Indicates the number of active alarms triggered from the applications.
- **FIS in progress** – Indicates the AWS FIS experiment that is currently running for your application.

To view the alarms implemented in your application

1. In the left navigation menu, choose **Applications**.
2. Select an application from the **Applications** table.
3. In the application summary page, the **Implemented alarms** table displays all the recommended alarms that are implemented in your application.

   To find a specific alarm in the **Implemented alarms** table, in the **Find alarms by text, property, or value** box, select one of the following fields, choose an operation, and then type a value.

   - **Alarm name** – Name of the alarm that you have set for your application.
   - **Description** – Describes the objective of the alarm.
   - **State** – Indicates the current implementation state of the Amazon CloudWatch alarm.

     This column displays one of the following values:

- **Implemented** – Indicates that the alarms recommended by AWS Resilience Hub are implemented in your application. Choose the number below to view all the recommended and implemented alarms in **Operational recommendations** tab.
- **Not implemented** – Indicates that the alarms recommended by AWS Resilience Hub are included but not implemented in your application. Choose the number below to view all the recommended and non-implemented alarms in **Operational recommendations** tab.
- **Excluded** – Indicates that the alarms recommended by AWS Resilience Hub are excluded from your application. Choose the number below to view all the recommended and excluded alarms in **Operational recommendations** tab. For more information about including and excluding recommended alarms, see Including or excluding operational recommendations.
- **Inactive** – Indicates that the alarms are deployed to Amazon CloudWatch, but the status is set to **INSUFFICIENT_DATA** in Amazon CloudWatch. Choose the number below to view all the implemented and inactive alarms in **Operational recommendations** tab.

- **Source template** – Provides the Amazon Resource Name (ARN) of the AWS CloudFormation stack that contains the alarm details.
- **Resource** – Displays the resources that this alarm is attached to and was implemented for.
- **Metric** – Displays the Amazon CloudWatch metric assigned for the alarm. For more information about Amazon CloudWatch metrics, see Amazon CloudWatch Metrics.
- **Last change** – Displays the date and time an alarm was last modified.

To view the recommended alarms from assessments

1. In the left navigation menu, choose **Applications**.
2. Select an application from the **Applications** table.

   To find an application, enter the application name in the **Find applications** box.
3. Choose **Assessments** tab.

   In **Resiliency assessments** table, you can identify your assessments using the following information:

   - **Name** – Name of the assessment you had provided at the time of creation.
   - **Status** – Indicates the execution state of the assessment.
   - **Compliance status** – Indicates if the assessment is compliant with the resiliency policy.

- **Resiliency drift status** – Indicates if your application has drifted or not from the previous successful assessment.
- **App version** – Version of your application.
- **Invoker** – Indicates the role that invoked the assessment.
- **Start time** – Indicates the start time of the assessment.
- **End time** – Indicates the end time of the assessment.
- **ARN** – The Amazon Resource Name (ARN) of the assessment.

4. Select an assessment from the **Resiliency assessments** table.
5. Choose **Operational recommendations** tab.
6. If not selected by default, choose **Alarms** tab.

   In **Alarms** table, you can identify the recommended alarms using the following:

   - **Name** – Name of the alarm that you have set for your application.
   - **Description** – Describes the objective of the alarm.
   - **State** – Indicates the current implementation state of the Amazon CloudWatch alarms.

     This column displays one of the following values:

     - **Implemented** – Indicates that the alarm is implemented in your application. Choosing the number below will filter the **Alarms** table to display all the recommended alarms that are implemented in your application.
     - **Not implemented** – Indicates that the alarm is not implemented or included in your application. Choosing the number below will filter the **Alarms** table to display all the recommended alarms that are not implemented in your application.
     - **Excluded** – Indicates that the alarm is excluded from the application. Choosing the number below will filter the **Alarms** table to display all the recommended alarms that are excluded from your application. For more information about including and excluding recommended alarms, see the section called “Including or excluding operational recommendations”.
     - **Inactive** – Indicates that the alarms are deployed to Amazon CloudWatch, but the status is set to **INSUFFICIENT_DATA** in Amazon CloudWatch. Choosing the number below will filter the **Alarms** table to display all the implemented and inactive alarms.
   - **Configuration** – Indicates if there are any pending configuration dependencies that needs to be addressed.

- **Type** – Indicates the type of alarm.
- **AppComponent** – Indicates the Application Components (AppComponents) that are associated with this alarm.
- **Reference ID** – Indicates the logical identifier of the AWS CloudFormation stack event in AWS CloudFormation.
- **Recommendation ID** – Indicates the logical identifier of the AWS CloudFormation stack resource in AWS CloudFormation.

Managing standard operating procedures

A standard operating procedure (SOP) is a prescriptive set of steps designed to efficiently recover your application in the event of an outage or alarm. Prepare, test, and measure your SOPs in advance to ensure timely recovery in the event of an operational outage.

Based on your Application Components, AWS Resilience Hub recommends the SOPs you should prepare. AWS Resilience Hub works with Systems Manager to automate the steps of your SOPs by providing a number of SSM documents you can use as the basis for those SOPs.

For example, AWS Resilience Hub may recommend an SOP for adding disk space based on an existing SSM Automation document. To run this SSM document, you require a specific IAM role with the correct permissions. AWS Resilience Hub creates metadata in your application indicating which SSM automation document to run in the case of disk shortage, and which IAM role is required to run that SSM document. This metadata is then saved in an SSM parameter.

In addition to configuring the SSM automation, it is also best practice to test it with an AWS FIS experiment. Therefore, AWS Resilience Hub also provides an AWS FIS experiment that calls the SSM automation document - this way, you can proactively test your application to make sure the SOP you've created does the intended job.

AWS Resilience Hub provides its recommendations in the form of an CloudFormation template you can add to your application code base. This template provides:

- The IAM role with the permissions required to run the SOP.
- An AWS FIS experiment you can use to test the SOP.
- An SSM parameter that contains application metadata indicating which SSM document and which IAM role is to be run as the SOP, and on which resource. For example: `$(DocumentName) for SOP $(HandleCrisisA) on $(ResourceA)`.

Creating an SOP may require some trial and error. Running a resiliency assessment against your application and generating an CloudFormation template from the AWS Resilience Hub recommendations is a good start. Use the CloudFormation template to generate an CloudFormation stack, then use the SSM parameters and their default values in your SOP. Run the SOP and see what refinements you need to make.

Because all applications have differing requirements, the default list of SSM documents that AWS Resilience Hub provides will not be sufficient for all of your needs. You can, however, copy the default SSM documents and use them as a basis to create your own custom documents tailored for your application. You can also create your own entirely new SSM documents. If you create your own SSM documents instead of modifying the defaults, you must associate them with SSM parameters, so the correct SSM document is called when the SOP runs.

When you've finalized your SOP by creating the necessary SSM documents and updating the parameter and document associations as necessary, add the SSM documents directly to your code base, and make any subsequent changes or customizations there. That way, every time you deploy your application, you'll also deploy the most up-to-date SOP.

Topics

- Building an SOP based on AWS Resilience Hub recommendations
- Creating a custom SSM document
- Using a custom SSM document instead of the default
- Testing SOPs
- Viewing standard operating procedures

Building an SOP based on AWS Resilience Hub recommendations

To build an SOP based on AWS Resilience Hub recommendations, you need an AWS Resilience Hub application with a resiliency policy attached to it, and you need to have run a resiliency assessment against that application. The resiliency assessment generates the recommendations for your SOP.

To build an SOP based on AWS Resilience Hub recommendations, you must create an CloudFormation template for the recommended SOPs and include them in your code base.

Create an CloudFormation template for the SOP recommendations

1. Open the AWS Resilience Hub console.
2. In the navigation pane, choose **Applications**.

3. From the list of applications, choose the application you want to create an SOP for.
4. Choose **Assessments** tab.
5. Select an assessment from the **Resiliency assessments** table. If you don't have an assessment, complete the procedure in the section called "Running resiliency assessments in AWS Resilience Hub" and then return to this step.
6. Under **Operational recommendations**, choose **Standard operating procedures**.
7. Select all the SOP recommendations you want to include.
8. Choose **Create CloudFormation template**. This can take up to a few minutes to create the AWS CloudFormation template.

   Complete the following procedure to include the SOP recommendations in your code base.

To include the AWS Resilience Hub recommendations in your code base

1. In **Operational recommendations**, choose **Templates**.
2. In the list of templates, choose the name of the SOP template you just created.

   You can identify the SOPs that are implemented in your application using the following information:

   - **SOP name** – Name of the SOP that you have defined for your application.
   - **Description** – Describes the objective of the SOP.
   - **SSM document** – Amazon S3 URL of the SSM document that contains the SOP definition.
   - **Test run** – Amazon S3 URL of the document that contains the results of the latest test.
   - **Source template** – Provides the Amazon Resource Name (ARN) of the AWS CloudFormation stack that contains the SOP details.
3. Under **Template details**, choose the link in **Templates S3 Path** to open the template object in Amazon S3 console.
4. In Amazon S3 console, from **Objects** table, choose the SOP folder link.
5. To copy the Amazon S3 path, select the check box in front of the JSON file and choose **Copy URL**.
6. Create an AWS CloudFormation stack from AWS CloudFormation console. For more information about creating an AWS CloudFormation stack, see https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-console-create-stack.html.

While creating the AWS CloudFormation stack, you must provide the Amazon S3 path that you copied from the previous step.

Creating a custom SSM document

To fully automate the recovery of your application, you may need to create a custom SSM document for your SOP in Systems Manager console. You can modify an existing SSM document as a base, or you can create a new SSM document.

For detailed information on using Systems Manager to create an SSM document, see Walkthrough: Using Document Builder to create a custom runbook.

For information about SSM document syntax, see SSM document syntax.

For information about automating SSM document actions, see Systems Manager automation actions reference.

Using a custom SSM document instead of the default

To replace the SSM document AWS Resilience Hub suggested for your SOP with a custom document you've created, work directly in your code base. In addition to adding your new custom SSM automation document, you'll also:

1. Add the IAM permissions required to run the automation.
2. Add an AWS FIS experiment to test your SSM document.
3. Add an SSM parameter that points to the automation document you want to use as the SOP.

Generally, it's most efficient to work with the suggested default values in AWS Resilience Hub and customize them as necessary. For example, add or remove permissions as necessary for the IAM role, change the AWS FIS experiment setup to point to the new SSM document, or change the SSM parameter to point to your new SSM document.

Testing SOPs

As previously mentioned, best practice is to add AWS FIS experiments to your CI/CD pipelines to test your SOPs regularly; this ensures they're ready to go if an outage occurs.

Test both AWS Resilience Hub-provided and custom SOPs.

Viewing standard operating procedures

To view the implemented SOPs from applications

1. In the left navigation menu, choose **Applications**.
2. In **Applications**, open an application.
3. Choose **Standard operating procedures** tab.

   In **Standard operating procedures summary** section, the **Implemented standard operating procedures** table displays the list of SOPs that are generated from SOP recommendations.

   You can identify your SOPs by the following:

   - **SOP name** – Name of the SOP that you have defined for your application.
   - **SSM document** – S3 URL of the Amazon EC2 Systems Manager document that contains the SOP definition.
   - **Description** – Describes the objective of the SOP.
   - **Test run** – S3 URL of the document that contains the results of the latest test.
   - **Reference ID** – Identifier of the referenced SOP recommendation.
   - **Resource ID** – Identifier of the resource for which the SOP recommendation is implemented.

To view the recommended SOPs from assessments

1. In the left navigation menu, choose **Applications**.
2. Select an application from the **Applications** table.

   To find an application, enter the application name in the **Find applications** box.
3. Choose **Assessments** tab.

   In **Resiliency assessments** table, you can identify your assessments using the following information:

   - **Name** – Name of the assessment you had provided at the time of creation.
   - **Status** – Indicates the execution state of the assessment.
   - **Compliance status** – Indicates if the assessment is compliant with the resiliency policy.
   - **Resiliency drift status** – Indicates if your application has drifted or not from the previous successful assessment.

- **App version** – Version of your application.
- **Invoker** – Indicates the role that invoked the assessment.
- **Start time** – Indicates the start time of the assessment.
- **End time** – Indicates the end time of the assessment.
- **ARN** – The Amazon Resource Name (ARN) of the assessment.

4. Select an assessment from the **Resiliency assessments** table.
5. Choose **Operational recommendations** tab.
6. Choose **Standard operating procedures** tab.

   In the **Standard operating procedures** table, you can understand more about the recommended SOPs using the following information:

   - **Name** – Name of the recommended SOP.
   - **Description** – Describes the objective of the SOP.
   - **State** – Indicates the current implementation state of the SOP. That is, **Implemented**, **Not implemented**, and **Excluded**.
   - **Configuration** – Indicates if there are any pending configuration dependencies that needs to be addressed.
   - **Type** – Indicates the type of SOP.
   - **AppComponent** – Indicates the Application Components (AppComponents) that are associated with this SOP. For more information about supported AppComponents, see Grouping resources in an AppComponent.
   - **Reference ID** – Indicates the logical identifier of the AWS CloudFormation stack event in AWS CloudFormation.
   - **Recommendation ID** – Indicates the logical identifier of the AWS CloudFormation stack resource in AWS CloudFormation.

Managing AWS Fault Injection Service experiments

This section describes how to manage AWS Fault Injection Service (AWS FIS) experiments in AWS Resilience Hub. You run AWS FIS experiments to measure the resiliency of your AWS resources and the amount of time it takes to recover from application, infrastructure, availability zone, and AWS Region incidents.

To measure resiliency, these AWS FIS experiments simulate disruptions to your AWS resources. Examples of disruptions include network unavailable errors, failovers, stopped processes on Amazon EC2 or AWS ASG, boot recovery in Amazon RDS, and problems with your Availability Zone. When the AWS FIS experiment concludes, you can estimate whether an application can recover from the outage types defined in the RTO target of the resiliency policy.

All the experiments in AWS Resilience Hub are built using AWS FIS and they execute AWS FIS actions. AWS FIS experiments use only AWS FIS automation actions that are customized to specific AWS services (such as Amazon EKS action). For more information about AWS FIS actions, see AWS FIS actions reference.

You can use the AWS FIS experiments in their default state or customize them based on your requirements. For more information about managing AWS FIS experiments from AWS Resilience Hub console and AWS FIS console, see the following topics:

- AWS Resilience Hub console
  - Viewing AWS FIS experiments
    - To view the list of implemented AWS FIS experiments from applications
    - To view the recommended AWS FIS experiments from assessments
  - the section called “Running AWS FIS experiments”
  - the section called “AWS Fault Injection Service experiment failures/status check”
- AWS FIS console
  - Managing your AWS FIS experiments
  - Working with the AWS FIS scenario library
  - Managing AWS FIS experiment templates

Initiating, creating, and running AWS FIS experiments

AWS Resilience Hub simplifies AWS FIS experiments by integrating with AWS FIS experiments. It provides tailored recommendations and allows initiating AWS FIS experiments with pre-populated templates mapped to your Application Components (AppComponents), enabling efficient resilience testing.

To initiate an AWS FIS experiment from Operational recommendations

1. Open the AWS Resilience Hub console.

2. In the navigation pane, choose **Applications**.
3. From the list of applications, choose the application you want to create a test for.
4. Choose **Assessments** tab.
5. Select an assessment from the **Resiliency assessments** table. If you don't have an assessment, complete the procedure in the section called "Running resiliency assessments in AWS Resilience Hub" and then return to this step.
6. Choose **Operational recommendations** tab.
7. Choose the right arrow before **Fault injection experiments**.

   This section lists all the AWS FIS experiments recommended by AWS Resilience Hub for your application to stress-test and improve its resilience. Based on your implementation, the AWS FIS experiments are categorized into the following states:

   - **Implemented** – Indicates that the experiments recommended by AWS Resilience Hub are implemented in your application. Choose the number below to view all the implemented experiments in the **Experiments** table.
   - **Partially implemented** – Indicates that the experiments recommended by AWS Resilience Hub are partially implemented in your application. Choose the number below to view all the partially implemented experiments in the **Experiments** table.
   - **Not implemented** – Indicates that the experiments recommended by AWS Resilience Hub are unimplemented in your application. Choose the number below to view all the unimplemented experiments in the **Experiments** table.
   - **Excluded** – Indicates that the experiments recommended by AWS Resilience Hub are excluded from your application. Choose the number below to view all the excluded experiments in the **Experiments** table. For more information about including and excluding recommended experiments, see Including or excluding operational recommendations.

   Experiments table lists all the implemented AWS FIS experiments that impact the resiliency score of your application. You can identify the AWS FIS experiments using the following information:

   - **Action name** – Indicates the AWS FIS action recommended for your application. Choose the action name to view all the recommended AppComponents on the **AWS FIS experiment details** page. When the **State** is set to **Not trackable**, it indicates that the AWS FIS experiment is a scenario. Choose the scenario name to view its details on the **Scenario library** page in the AWS FIS console.

- **State** – Indicates the current implementation state of the AWS FIS experiment. That is, **Implemented**, **Partially implemented**, **Not implemented**, and **Excluded**.

  > **Note**
  >
  > AWS FIS scenario is a console-only feature with multiple predefined actions. Hence, AWS Resilience Hub cannot track it and it will set the **State** to **Not trackable**.

- **Description** – Describes the objective of the AWS FIS action.

8. Select an AWS FIS action for which you want to initiate an experiment.

   In the AWS FIS experiment recommendation section, you can understand more about the experiments you need implement on the AppComponents using the following information:

   - **Name** – Name of the AppComponent in which the resources are grouped into.
   - **State** – Indicates the current implementation state of the AWS FIS action. That is, **Implemented**, **Partially implemented**, **Not implemented**, and **Excluded**.

     > **Note**
     >
     > AWS FIS scenario is a console-only feature with multiple predefined actions. Hence, AWS Resilience Hub cannot track it and it will set the **State** to **Not trackable**.

   - **Target selection** – Indicates how the resources will be included in the experiment when you choose **Initiate experiment**. If AWS Resilience Hub doesn't automatically determine target resources, hover over the respective **Target selection** field for guidance on adding them.
   - **Resources** – Indicates the number of resources grouped under the AppComponent. Choose the number to view these resources in the **Resources** dialog box. You can identify the resources using the following:
     - **Logical ID** – Indicates the logical ID of the resource. A logical ID is a name used to identify resources in your AWS CloudFormation, Terraform state file, myApplications application, AWS Resource Groups resource, or Amazon Elastic Kubernetes Service cluster.
     - **Physical ID** – Indicates the actual assigned identifier for the resource, such as an Amazon EC2 instance ID or an Amazon S3 bucket name.
     - **Type** – Indicates the type of resource.
     - **Region** – Indicates the AWS Region in which the resource is located.

9. Select an AppComponent and choose **Include** or **Exclude** to include or exclude the AppComponent in the AWS FIS experiment, respectively.
10. Choose **Initiate experiment**.

    AWS Resilience Hub will redirect you to **Specify template details** page in the AWS FIS console, opening it in a new tab.
11. To create an experiment template, complete the steps in To create an experiment template using the console.

    Additionally, after you enter the template details and choose **Next** in the **Specify template details** page of the AWS FIS console by following the steps in To create an experiment template using the console, AWS Resilience Hub automatically tries to map **Actions** and **Targets** for your resource types in the **Actions and targets** page. However, to improve the coverage, you can manually add actions and targets by choosing **Add action** and **Add target**, respectively, and complete the rest of the procedure to create your experiment.

Running AWS FIS experiments

After creating an experiment in AWS FIS console, follow the steps in Start an experiment from a template to run an experiment in AWS FIS console. If you want AWS Resilience Hub to detect the latest experiments you have run in AWS FIS, you must run a new assessment. For more information about running assessments, see Running resiliency assessments in AWS Resilience Hub.

Viewing AWS FIS experiments

In AWS Resilience Hub, view the AWS FIS experiments that you set up to measure the resiliency of your AWS resources and the amount of time it takes to recover from application, infrastructure, availability zone, and AWS Region incidents.

To view the list of active AWS FIS experiments from the dashboard, choose **Dashboard** from the left navigation menu.

In the **Implemented experiments** table, you can identify the AWS FIS experiments using the following information:

- **Experiment ID** – Identifier of the AWS FIS experiment.
- **Action** – Indicates the AWS FIS action associated with the AWS FIS experiment. Additionally, if there are more than one action, it highlights the number of AWS FIS actions associated with

the AWS FIS experiment. You can identify the details by hovering over them or by navigating to them.

- **Experiment template ID** – Identifier of the AWS FIS experiment template that was used to create the AWS FIS experiment.

To view the list of implemented AWS FIS experiments from applications

1. In the left navigation menu, choose **Applications**.
2. Select an application from the **Applications** table.

   To find an application, enter the application name in the **Find applications** box.
3. Choose **Fault injection experiments**.

   In the **Implemented experiments** table, you can identify the AWS FIS experiments implemented in your application using the following information:

   - **Experiment ID** – Identifier of the AWS FIS experiment.
   - **Action** – Indicates the AWS FIS action associated with the AWS FIS experiment. Additionally, if there are more than one action, it highlights the number of AWS FIS actions associated with the AWS FIS experiment. You can identify the details by hovering over them or by navigating to them.
   - **Experiment template ID** – Identifier of the AWS FIS experiment template that was used to create the AWS FIS experiment.

To view the recommended AWS FIS experiments from assessments

1. In the left navigation menu, choose **Applications**.
2. Select an application from the **Applications** table.

   To find an application, enter the application name in the **Find applications** box.
3. Choose **Assessments** tab.

   In the **Assessments** table, you can identify your assessments using the following information:

   - **Name** – Name of the assessment you had provided at the time of creation.
   - **Status** – Indicates the execution state of the assessment.
   - **Compliance status** – Indicates if the assessment is compliant with the resiliency policy.

- **Resiliency** – Indicates if your application has drifted from the RTO and RPO targets defined in the attached resiliency policy or not from the previous successful assessment.
- **App version** – Version of your application that was assessed.
- **Invoker** – Indicates the role that invoked the assessment.
- **Start time** – Indicates the start time of the assessment.
- **End time** – Indicates the end time of the assessment.
- **ARN** – The Amazon Resource Name (ARN) of the assessment.

4. Select an assessment from the **Assessments** table.
5. Choose **Operational recommendations**.
6. Choose the right arrow before **Fault injection experiments**.

   This section lists all the AWS FIS experiments recommended by AWS Resilience Hub for your application to stress-test and improve its resilience. Based on your implementation, the AWS FIS experiments are categorized into the following states:

   - **Implemented** – Indicates that the experiments recommended by AWS Resilience Hub are implemented in your application. Choose the number below to view all the implemented experiments in the **Experiments** table.
   - **Partially implemented** – Indicates that the experiments recommended by AWS Resilience Hub are partially implemented in your application. Choose the number below to view all the partially implemented experiments in the **Experiments** table.
   - **Not implemented** – Indicates that the experiments recommended by AWS Resilience Hub are unimplemented in your application. Choose the number below to view all the unimplemented experiments in the **Experiments** table.
   - **Excluded** – Indicates that the experiments recommended by AWS Resilience Hub are excluded from your application. Choose the number below to view all the excluded experiments in the **Experiments** table. For more information about including and excluding recommended experiments, see Including or excluding operational recommendations.

   Experiments table lists all the implemented AWS FIS experiments that impact the resiliency score of your application. You can identify the AWS FIS experiments using the following information:

- **Action name** – Indicates the AWS FIS action recommended for your application. When the **State** is set to **Not trackable**, it indicates that the AWS FIS experiment is a scenario. Choose the scenario name to view its details on the **Scenario library** page in the AWS FIS console.
- **State** – Indicates the current implementation state of the AWS FIS experiment. That is, **Implemented**, **Partially implemented**, **Not implemented**, and **Excluded**.

  > **Note**
  >
  > AWS FIS scenario is a console-only feature with multiple predefined actions. Hence, AWS Resilience Hub cannot track it and it will set the **State** to **Not trackable**.

- **Description** – Describes the objective of the AWS FIS action.

AWS Fault Injection Service experiment failures/status check

AWS Resilience Hub allows you to track the status of your experiment that you have started. For more information, see the To view the recommended AWS FIS experiments from assessments procedure.

Topics

- Analyzing AWS FIS experiment execution using AWS Systems Manager
- AWS FIS experiment failures while testing Kubernetes pods running in your Amazon Elastic Kubernetes Service clusters

Analyzing AWS FIS experiment execution using AWS Systems Manager

After running an AWS FIS experiment, you can view the execution details in the AWS Systems Manager.

1. Go to **CloudTrail** > **Event History**.
2. Filter events by **User name** using the experiment ID.
3. View the StartAutomationExecution entry. **Request ID** is the SSM automation ID.
4. Go to **AWS Systems Manager** > **Automation**.
5. Filter by **Execution ID** using SSM automation ID and view the automation details.

You can analyze the execution with any Systems Manager automation. For more information, see the AWS Systems Manager Automation user guide. The execution input parameters appear in the **Input parameters** section of the **Execution Detail** and include optional parameters not appearing in the AWS FIS experiment.

You can find information on step status and other step details by drilling down to specific steps within the Execution steps.

Common failures

The following are common failures encountered while executing an assessment report:

- Alarm template was not deployed before the Test/SOP experiment was executed. This causes an error message during the automation step.
  - **Failure message:** `The following parameters were not found: [/ResilienceHub/Alarm/3dee49a1-9877-452a-bb0c-a958479a8ef2/nat-gw-alarm-bytes-out-to-source-2020-09-21_nat-02ad9bc4fbd4e6135]. Make sure all the SSM parameters in automation document are created in SSM Parameter Store.`
  - **Remediation:** Ensure to render the relevant alarm and deploy the resulting template before rerunning the fault injection experiment.
- Missing permissions in the execution role. This error message occurs if the provided execution role is missing a permission and appears within the step details.
  - **Failure message:** `An error occurred (Unauthorized Operation) when calling the DescribeInstanceStatus operation: You are not authorized to perform this operation. Please Refer to Automation Service Troubleshooting Guide for more diagnosis details.`
  - **Remediation**: Verify you provided the correct execution role. If this was done, add the required permission and rerun the assessment.
- Execution succeeded but did not have the expected result. This is the result of incorrect parameters or an internal automation issue.
  - **Failure message:** The execution succeeded, so no error message is shown.
  - **Remediation:** Check the input parameters and look at the executed steps as explained in the Analyze AWS FIS experiment execution before examining the individual steps for expected inputs and outputs.

AWS FIS experiment failures while testing Kubernetes pods running in your Amazon Elastic Kubernetes Service clusters

The following are common Amazon Elastic Kubernetes Service (Amazon EKS) failures encountered while testing Kubernetes pods running in your Amazon EKS clusters:

- Incorrect configuration of IAM roles for AWS FIS experiments or the Kubernetes service account.
  - **Failure messages:**
    - `Error resolving targets. Kubernetes API returned ApiException with error code 401.`
    - `Error resolving targets. Kubernetes API returned ApiException with error code 403.`
    - `Unable to inject AWS FIS Pod: Kubernetes API returned status code 403. Check Amazon EKS logs for more details.`
  - **Remediation:** Verify the following.
    - Ensure that you have followed the instruction in Use the AWS FIS`aws:eks:pod` actions.
    - Ensure that you have created and configured a Kubernetes Service Account with the necessary RBAC permissions and the correct namespace.
    - Ensure that you have mapped the provided IAM role (see the output of the CloudFormation stack of the test) to the Kubernetes user.
- Unable to start AWS FIS Pod: Max failed sidecar containers reached. This usually happens when the memory is not sufficient to run the AWS FIS sidecar container.
  - **Failure message:** `Unable to heartbeat FIS Pod: Max failed sidecar containers reached.`
  - **Remediation:** One option to avoid this error is to reduce the target load percentage to be aligned with the available memory or CPU.
- Alarm assertion failed at the beginning of the experiment. This error occurs because the related alarm has no datapoint.
  - **Failure message:** `Assertion failed for the following alarms`. Lists all the alarms for which the assertion has failed.
  - **Remediation:** Ensure that Container Insights are correctly installed for the alarms and the alarm is not turned on (in `ALARM` state).

Understanding resiliency scores

This section describes how AWS Resilience Hub quantifies application readiness from different disruption scenarios.

AWS Resilience Hub provides resiliency score that represents the resiliency posture of the application. This score reflects how closely the application follows our recommendations for meeting the application's resiliency policy, alarms, standard operating procedures (SOPs), and tests. Based on the type of resources the application uses, AWS Resilience Hub recommends alarms, SOPs, and a set of tests for each disruption type.

The top resiliency score is 100 points. To achieve the best possible score or the top score, you must implement all the recommended alarms, SOPs, and tests in your application. For example, AWS Resilience Hub recommends one test with one alarm and one SOP. The test runs and fires the alarm and initiates the associated SOP. If they perform successfully and if the application meets the resiliency policy, it receives a resiliency score close to or equal to 100 points.

After running first assessment, AWS Resilience Hub provides an option to exclude operational recommendations from your application. To understand the impact of the excluded recommendations on the resiliency score, you must run a new assessment. However, you can always include the excluded recommendations in your application and run a new assessment. For more information about including and excluding alarm, SOP, and test recommendations, see the section called “Including or excluding operational recommendations”.

Accessing the Resiliency score of your applications

You can view the Resiliency score of your application by choosing **Dashboard** or **Applications** from the navigation menu.

Accessing the Resiliency score from Dashboard

1. In the left navigation menu, choose **Dashboard**.
2. In **Application resiliency score over time**, choose one or more applications in the **Choose up to 4 applications** dropdown list.
3. The **Resiliency score** chart displays the resiliency score for all the chosen applications.

Accessing the Resiliency score from Applications

1. In the left navigation menu, choose **Applications**.

2. In **Applications**, open an application.
3. Choose **Summary**.

   The **Resiliency score** chart displays the trend of your application's resiliency score for up to one year. AWS Resilience Hub displays action items, resiliency policy breaches, and operational recommendations that need to be addressed for improving and achieving the maximum possible resiliency score using the following:

   - To view the action items that need to be completed for improving and achieving the maximum possible resiliency score, choose **Action items** tab. When selected, AWS Resilience Hub displays the following:
     - **RTO/RPO** – Indicates the number of recovery times (RTO/RPOs) that need to be fixed to resolve the breaches in your application's resiliency policy. Choose the value to view the RTO/RPO details in the assessment report of your application.
     - **Alarms** – Indicates the number of recommended Amazon CloudWatch alarms that need to be implemented in your application. Choose the value to view the Amazon CloudWatch alarms that need to be fixed in the assessment report of your application.
     - **SOPs** – Indicates the number of recommended SOPs that need to be implemented in your application. Choose the value to view the SOPs that need to be fixed in the assessment report of your application.
     - **FIS** – Indicates the number of recommended tests that need to be implemented in your application. Choose the value to view the tests that need to be fixed in the assessment report of your application.
   - To view the score of each component that affects your resiliency score, choose **Score breakdown**. When selected, AWS Resilience Hub displays the following:
     - **RTO/RPO compliance** – Indicates how compliant the Applications Components (AppComponents) are with the estimated workload recovery times, and the target recovery times that are defined in your application’s resiliency policy. Choose the value to view the RTO/RPO estimations in the assessment report of your application.
     - **Alarms implemented** – Indicates the actual contribution of the implemented Amazon CloudWatch alarms compared to its maximum possible contribution towards the resiliency score of your application. Choose the value to view the implemented Amazon CloudWatch alarms in the assessment report of your application.
     - **SOPs implemented** – Indicates the actual contribution of the implemented SOPs compared to its maximum possible contribution towards the resiliency score of your

application. Choose the value to view the implemented SOPs in the assessment report of your application.

- **FIS experiments implemented** – Indicates the actual contribution of the implemented tests compared to its maximum possible contribution towards the resiliency score of your application. Choose the value to view the implemented tests in the assessment report of your application.

- To view the resiliency policy breaches and operational recommendations, choose the right arrow to expand the **Policy breach and operational recommendations breakdown** section. When expanded, AWS Resilience Hub displays the following:
  - **Resiliency policy breaches** – Indicates the number of Application Components that breaches your application's resiliency policy. Choose the value next to **RTO/RPO** to view the details in the **Resiliency recommendations** tab of your application's assessment report.
  - **Operational recommendations** – Indicates the operational recommendations that have not been implemented or executed to enhance the resiliency of your application using **Outstanding** and **Excluded** tabs. Operational recommendations include all the recommendations that are inactive and the ones that have not been implemented.

    To view the operational recommendations that need to be implemented, choose **Outstanding** tab. When selected, AWS Resilience Hub displays the following:

    - **Alarms** – Indicates the number of recommended Amazon CloudWatch alarms that need to be implemented.
    - **SOPs** – Indicates the number of recommended SOPs that need to be implemented.
    - **FIS** – Indicates the number of recommended tests that need to be implemented.

    To view the operational recommendations that are excluded from your application, choose **Excluded** tab. When selected AWS Resilience Hub displays the following:

    - **Alarms** – Indicates the number of recommended Amazon CloudWatch alarms that are excluded from your application.
    - **SOPs** – Indicates the number of recommended SOPs that are excluded from your application.
    - **FIS** – Indicates the number of recommended tests that are excluded from your application.

Calculating resiliency scores

The tables in this section explains the formulas used by AWS Resilience Hub to determine the scoring components of each recommendation type and the resiliency score of your application. All the resultant values determined by AWS Resilience Hub for scoring components of each recommendation type and the resiliency score of your application are rounded to their nearest point. For example, if two out of three alarms were implemented, the score would be 13.33 ((2/3) * 20) points. This value will be rounded to 13 points. For more information about weights used in the formulas within the tables, see the section called “Weights of AppComponents and disruption types” section.

Some of the scoring components can be obtained only through the `ScoringComponentResiliencyScore` API. For more information about this API, see ScoringComponentResiliencyScore.

Tables

- **Formulas to calculate the scoring component of each recommendation type**
- **Formula to calculate the resiliency score**
- **Formulas to calculate resiliency score for AppComponents and disruption types**

The following table explains the formulas used by AWS Resilience Hub to calculate the scoring component of each recommendation type.

Formulas to calculate the scoring component of each recommendation type

Scoring component	Description	Formula	Example
Test coverage (T)	A normalized score (0 -100 points) based on the number of tests that were successfully implemented and excluded, out of the total number of AWS Resilience Hub recommended tests.	`T = ((Total number of tests implemented) + (Total number of tests excluded)) / (Total number of tests recommended)` Parts of the formula are as follows:	If you have implemented 10 and excluded 5 tests out of 20 AWS Resilience Hub recommended tests, the test coverage

Scoring component	Description	Formula	Example
	Note To calculate the resiliency score, the recommend ed tests must have run successfully in the last 30 days for AWS Resilience Hub to consider it as implemented.	• **Total number of tests configured** – Indicates the total number of tests configured when the AWS CloudForm ation template is created and uploaded in the AWS CloudForm ation console. • **Total number of tests recommended** – Indicates the tests recommended by AWS Resilience Hub based on the application resources. • **Total number of tests excluded** – Indicates the number of recommend ed tests you have excluded from the application.	is calculated as follows: `T = (10 + 5) / 20` That is, `T = .75 or 75 points`

Scoring component	Description	Formula	Example
Alarms coverage (A)	A normalized score (0 -100 points) based on the number of Amazon CloudWatch alarms that were successfully implemented and excluded, out of the total number of AWS Resilience Hub recommended Amazon CloudWatch alarms. **Note** To calculate the resiliency score, the recommended alarms should be in **Ready** state for AWS Resilience Hub to consider it as implemented.	`A = ((Total number of alarms implemented) + (Total number of alarms excluded)) / (Total number of alarms recommended)` Parts of the formula are as follows: • **Total number of alarms configured** – Indicates the total number of Amazon CloudWatch alarms configured when the AWS CloudFormation template is created and uploaded in the AWS CloudFormation console. • **Total number of alarms recommended** – Indicates the Amazon CloudWatch alarms recommended by AWS Resilience Hub based on the application resources. • **Total number of alarms excluded** – Indicates the number of recommended Amazon CloudWatch alarms that you have	If you have implemented 10 and excluded 5 Amazon CloudWatch alarms out of 20 AWS Resilience Hub recommended Amazon CloudWatch alarms, the Amazon CloudWatch alarms coverage is calculated as follows: `A = (10 + 5) / 20` That is, `A = .75 or 75 points`

Scoring component	Description	Formula	Example
		excluded from the application.	

Scoring component	Description	Formula	Example
SOP coverage (S)	A normalized score (0 -100 points) based on the number of SOPs that were successfully implemented and excluded, out of the total number of AWS Resilience Hub recommended SOPs.	`S = ((Total number of SOPs implemented) + (Total number of SOPs excluded) ) / (Total number of SOPs recommended)` Parts of the formula are as follows: • **Total number of SOPs configured** – Indicates the total number of SOPs configured when the AWS CloudFormation template is created and uploaded in the AWS CloudFormation console. • **Total number of SOPs recommended** – Indicates the SOPs recommended by AWS Resilience Hub based on the application resources. • **Total number of SOPs excluded** – Indicates the number of recommended SOPs you have excluded from the application.	If you have implemented 10 and excluded 5 SOPs out of 20 AWS Resilience Hub recommended SOPs, the SOP coverage is calculated as follows: `S = (10 + 5) / 20` That is, `S = .75 or 75 points`

Scoring component	Description	Formula	Example

RTO/RPO compliance (P)	A normalized score (0 -100 points) based on the application meeting its resiliency policy.	`P = Total weights of disruption types meeting the application's resiliency policy / Total weights of all disruption types .`	If you application resiliency policy meets only for Availability Zone (AZ) and Infrastructure disruption types, the resiliency policy score (P) is calculated as follows:
			• If you have set Regional RTO and RPO targets, P is calculated as follows:
			`P = (20 + 30)/ 100`
			That is, `P = .5 or 50 points`
			• If you have not set Regional RTO and RPO targets, P is calculated as follows:
			`P = (22.22 + 33.33)/ 99.9`
			That is, `P = .55 or 55 points`

The following table explains the formula used by AWS Resilience Hub to calculate the resiliency score for your entire application.

Formula to calculate the Resiliency score

Scoring component	Description	Formula	Example
Resilienc y score for application (RS)	A normalized resilienc y score (0 -100 points) based on your applicati on meeting its resilienc y policy. Resiliency score per application is the weighted average of all the recommendation types. That is: `RS = Weighted Average (T, A, S, P)`	Resiliency score per application is calculate d using the following formula: `RS = (T * Weight(T) +` `A * Weight(A) +` `S * Weight(S) +` `P * Weight(P)) /` `(Weight(T) + Weight(A) + Weight(S) + Weight(P))`	Formulas to calculate the coverage of each recommendation type table are as follows: • `Test coverage (T) = .75` • `Alarms (A) = .75` • `SOPs (S) = .75` • `Meeting resiliency policy (P) = .5` The resiliency score per application is calculated as follows: `RS = ((.75 * .2) + (.75 * .2) + (.75 * .2) + (.5 * .4)) /(.2`

Scoring component	Description	Formula	Example
			+ .2 + .2 + .4) That is, RS = .65 or 65 points

The following table explains the formulas used by AWS Resilience Hub to calculate the resiliency score for Application Components (AppComponents) and disruption types. However, you can obtain the resiliency score of AppComponents and disruption types only through the following AWS Resilience Hub APIs:

- DescribeAppAssessment to obtain RSo
- ListAppComponentCompliances to obtain RSao and RSA

Formulas to calculate resiliency score for AppComponents and disruption types

Scoring component	Description	Formula	Example
Resiliency score per AppComponent and per disruption type (RSao)	A normalized score (0-100 points) based on the AppComponent meeting its resiliency policy per disruption type. Resiliency score per AppComponent and per disruption type is the weighted	The resiliency score per AppComponent and per disruption type is calculated using the following formula: RSao = (T * Weight(T) + A * Weight(A) + S * Weight(S) + P * Weight(P)) / (Weight(T) + Weight(A) + Weight(S) + Weight(P))	RSao assumptions for all the recommendation types are as follows: • Test coverage (T) = .75 • Alarms (A) = .75 • SOPs (S) = .75 • Meeting resiliency policy (P) = .5

Scoring component	Description	Formula	Example
	average of all the recommendation types. That is: `RSao = Weighted Average (T, A, S, P)` The values for `T, A, S, P` are calculated for all the recommended tests, alarms, SOPs, and meeting resiliency policy of the AppComponent and the disruption type.		The resiliency score per AppComponent and disruption type is calculated as follows: `RSao = ((.75 * .2) + (.75 * .2) + (.75 * .2) + (.5 * .4)) / (.2 + .2 + .2 + .4)` That is, `RSao = .65 or 65 points`

Scoring component	Description	Formula	Example
Resiliency score per AppComponent (RSa)	A normalized score (0 -100 points) based on meeting its resiliency policy. Resiliency score per AppComponent is the weighted average of all the recommendation types. That is: RSa = Weighted Average (T, A, S, P) The values for T, A, S, P are calculated for all the recommended tests, alarms, SOPs, and meeting resiliency policy of the AppComponent.	The resiliency score per AppComponent is calculated using the following formula: RSa = (T * Weight(T) + A * Weight(A) + S * Weight(S) + P * Weight(P)) / (Weight(T) + Weight(A) + Weight(S) + Weight(P))	RSa assumptions for all the recommendation types are as follows: • Test coverage (T) = .75 • Alarms (A) = .75 • SOPs (S) = .75 • Meeting resiliency policy (P) = .5 The resiliency score per AppComponent is calculated as follows: RSa = ((.75 * .2) + (.75 * .2) + (.75 * .2) + (.5 * .4)) / (.2 + .2 + .2 + .4) That is, RSa = .65 or 65 points

Scoring component	Description	Formula	Example
Resiliency score per disruption type (RSo)	A normalized score (0 -100 points) based on meeting its resilienc y policy. Resiliency score per disruption type is the weighted average of all the recommend ation types. That is: RSo = Weighted Average (T, A, S, P) The values for T, A, S, P are calculate d for all the recommend ed tests, alarms, SOPs, and meeting resiliency policy of the disruption type.	The resiliency score per disruptio n type is calculated using the following formula: RSo = (T * Weight(T) + A * Weight(A) + S * Weight(S) + P * Weight(P)) / (Weight(T) + Weight(A) + Weight(S) + Weight(P))	RSo assumptions for all the recommend ation types are as follows: • Test coverage (T) = .75 • Alarms (A) = .75 • SOPs (S) = .75 • Meeting resiliency policy (P) = .5 The resiliency score per disruption type is calculated as follows: RSo = ((.75 * .2) + (.75 * .2) + (.75 * .2) + (.5 *.4)) / (.2 + .2 + .2 + .4) That is, RSo = .65 or 65 points

Weights

AWS Resilience Hub assigns a weight to each recommendation type for the total resiliency score.

The following tables show the weight for alarms, SOPs, tests, meeting resiliency policy, and disruption types. Disruptions type include Application, Infrastructure, AZ, and Region.

> **Note**
>
> If you choose not to define Regional RTO or RPO targets for your policy, the weights for the other disruption types are increased accordingly as shown in **Weight when Region is not defined** column.

Weights for alarms, SOPs, tests, policy target

Recommendation type	Weight
Alarms	20 points
SOPs	20 points
Tests	20 points
Meeting resiliency policy	40 points

Weights for disruption type

Disruption type	Weight when Region is defined	Weight when Region is not defined
Application	40 points	44.44 points
Infrastructure	30 points	33.33 points
Availability Zone	20 points	22.22 points
Region	10 points	N/A

Integrating operational recommendations into your application with CloudFormation

After you choose **Create CloudFormation template** in the **Operational recommendations** page, AWS Resilience Hub creates an CloudFormation template that describes the specific alarm, standard operating procedure (SOP), or AWS FIS experiment for your application. The CloudFormation template is stored in an Amazon S3 bucket, and you can check the S3 path to the template in the **Template details** tab on the **Operational recommendations** page.

For example, the listing below shows a JSON-formatted CloudFormation template that describes an alarm recommendation rendered by AWS Resilience Hub. It's a Read Throttling Alarm for a DynamoDB table called `Employees`.

The `Resources` section of the template describes the `AWS::CloudWatch::Alarm` alarm that's activated when the number of read throttle events for the DynamoDB table exceeds 1. And the two `AWS::SSM::Parameter` resources define metadata that allow AWS Resilience Hub to identify installed resources without having to scan the actual application.

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Parameters" : {
    "SNSTopicARN" : {
      "Type" : "String",
      "Description" : "The ARN of the Amazon SNS topic to which alarm status changes
 are to be sent. This must be in the same Region being deployed.",
      "AllowedPattern" : "^arn:(aws|aws-cn|aws-iso|aws-iso-[a-z]{1}|aws-us-gov):sns:
([a-z]{2}-((iso[a-z]{0,1}-)|(gov-)){0,1}[a-z]+-[0-9]):[0-9]{12}:[A-Za-z0-9/][A-Za-
z0-9:_/+=,@.-]{1,256}$"
    }
  },
  "Resources" : {

 "ReadthrottleeventsthresholdexceededEmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9Alarm" :
 {
      "Type" : "AWS::CloudWatch::Alarm",
      "Properties" : {
        "AlarmDescription" : "An Alarm by AWS Resilience Hub that alerts when the
 number of read-throttle events are greater than 1.",
        "AlarmName" : "ResilienceHub-ReadThrottleEventsAlarm-2020-04-01_Employees-ON-
DEMAND-0-DynamoDBTable-PXBZQYH3DCJ9",
        "AlarmActions" : [ {
```

```
          "Ref" : "SNSTopicARN"
        } ],
        "MetricName" : "ReadThrottleEvents",
        "Namespace" : "AWS/DynamoDB",
        "Statistic" : "Sum",
        "Dimensions" : [ {
          "Name" : "TableName",
          "Value" : "Employees-ON-DEMAND-0-DynamoDBTable-PXBZQYH3DCJ9"
        } ],
        "Period" : 60,
        "EvaluationPeriods" : 1,
        "DatapointsToAlarm" : 1,
        "Threshold" : 1,
        "ComparisonOperator" : "GreaterThanOrEqualToThreshold",
        "TreatMissingData" : "notBreaching",
        "Unit" : "Count"
      },
      "Metadata" : {
        "AWS::ResilienceHub::Monitoring" : {
          "recommendationId" : "dynamodb:alarm:health-read_throttle_events:2020-04-01"
        }
      }
    },

 "dynamodbalarmhealthreadthrottleevents20200401EmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9Alarm
 {
      "Type" : "AWS::SSM::Parameter",
      "Properties" : {
        "Name" : "/ResilienceHub/Alarm/3f904525-4bfa-430f-96ef-58ec9b19aa73/dynamodb-
alarm-health-read-throttle-events-2020-04-01_Employees-ON-DEMAND-0-DynamoDBTable-
PXBZQYH3DCJ9",
        "Type" : "String",
        "Value" : {
          "Fn::Sub" :
 "${ReadthrottleeventsthresholdexceededEmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9Alarm}"
        },
        "Description" : "SSM Parameter for identifying installed resources."
      }
    },

 "dynamodbalarmhealthreadthrottleevents20200401EmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9Alarm
 {
      "Type" : "AWS::SSM::Parameter",
      "Properties" : {
```

```
        "Name" : "/ResilienceHub/Info/Alarm/3f904525-4bfa-430f-96ef-58ec9b19aa73/
dynamodb-alarm-health-read-throttle-events-2020-04-01_Employees-ON-DEMAND-0-
DynamoDBTable-PXBZQYH3DCJ9",
        "Type" : "String",
        "Value" : {
          "Fn::Sub" : "{\"alarmName\":
\"${ReadthrottleeventsthresholdexceededEmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9Alarm}\",
\"referenceId\":\"dynamodb:alarm:health_read_throttle_events:2020-04-01\",
\"resourceId\":\"Employees-ON-DEMAND-0-DynamoDBTable-PXBZQYH3DCJ9\",\"relatedSOPs\":
[\"dynamodb:sop:update_provisioned_capacity:2020-04-01\"]}"
        },
        "Description" : "SSM Parameter for identifying installed resources."
      }
    }
  }
}
```

Modifying the CloudFormation template

The easiest way to integrate an alarm, SOP, or AWS FIS resource into your main application is to simply add it as another resource in the template that describes your application template. The JSON-formatted file provided below provides a basic outline of how a DynamoDB table is described in an CloudFormation template. A real application is likely to include several more resources, such as additional tables.

```
{
   "AWSTemplateFormatVersion": "2010-09-09T00:00:00.000Z",
   "Description": "Application Stack with Employees Table",
   "Outputs": {
      "DynamoDBTable": {
         "Description": "The DynamoDB Table Name",
         "Value": {"Ref": "Employees"}
      }
   },
   "Resources": {
      "Employees": {
         "Type": "AWS::DynamoDB::Table",
         "Properties": {
            "BillingMode": "PAY_PER_REQUEST",
            "AttributeDefinitions": [
               {
                  "AttributeName": "USER_ID",
                  "AttributeType": "S"
```

```
            },
            {
               "AttributeName": "RANGE_ATTRIBUTE",
               "AttributeType": "S"
            }
         ],
         "KeySchema": [
            {
               "AttributeName": "USER_ID",
               "KeyType": "HASH"
            },
            {
               "AttributeName": "RANGE_ATTRIBUTE",
               "KeyType": "RANGE"
            }
         ],
         "PointInTimeRecoverySpecification": {
            "PointInTimeRecoveryEnabled": true
         },
         "Tags": [
            {
               "Key": "Key",
               "Value": "Value"
            }
         ],
         "LocalSecondaryIndexes": [
            {
               "IndexName": "resiliencehub-index-local-1",
               "KeySchema": [
                  {
                     "AttributeName": "USER_ID",
                     "KeyType": "HASH"
                  },
                  {
                     "AttributeName": "RANGE_ATTRIBUTE",
                     "KeyType": "RANGE"
                  }
               ],
               "Projection": {
                  "ProjectionType": "ALL"
               }
            }
         ],
         "GlobalSecondaryIndexes": [
```

```
                {
                   "IndexName": "resiliencehub-index-1",
                   "KeySchema": [
                      {
                         "AttributeName": "USER_ID",
                         "KeyType": "HASH"
                      }
                   ],
                   "Projection": {
                      "ProjectionType": "ALL"
                   }
                }
             ]
          }
       }
    }
}
```

To allow the alarm resource to be deployed with your application, you now need to replace the hardcoded resources with a dynamic reference in the application stacks.

So, in the `AWS::CloudWatch::Alarm` resource definition, change the following:

```
"Value" : "Employees-ON-DEMAND-0-DynamoDBTable-PXBZQYH3DCJ9"
```

to the below:

```
"Value" : {"Ref": "Employees"}
```

And under in the `AWS::SSM::Parameter` resource definition, change the following:

```
"Fn::Sub" : "{\"alarmName\":
\"${ReadthrottleeventsthresholdexceededDynamoDBEmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9Alarm
\"referenceId\":\"dynamodb:alarm:health_read_throttle_events:2020-04-01\",
\"resourceId\":\"Employees-ON-DEMAND-0-DynamoDBTable-PXBZQYH3DCJ9\",\"relatedSOPs\":
[\"dynamodb:sop:update_provisioned_capacity:2020-04-01\"]}"
```

to the below:

```
"Fn::Sub" : "{\"alarmName\":
\"${ReadthrottleeventsthresholdexceededEmployeesONDEMAND0DynamoDBTablePXBZQYH3DCJ9Alarm}\",
```

```
\"referenceId\":\"dynamodb:alarm:health_read_throttle_events:2020-04-01\",\"resourceId
\":\"${Employees}\",\"relatedSOPs\":
[\"dynamodb:sop:update_provisioned_capacity:2020-04-01\"]}"
```

When modifying CloudFormation templates for SOPs and AWS FIS experiments, you will take the same approach, replacing hardcoded reference IDs with dynamic references that continue to work even after hardware changes.

By using a reference to the DynamoDB table, you allow CloudFormation to do the following:

- Create the database table first.
- Always use the actual ID of the generated resource in the alarm, and update the alarm dynamically if CloudFormation needs to replace the resource.

> **Note**
>
> You can choose more advanced methods for managing your application resources with CloudFormation such as nesting stacks or referring to resource outputs in a separate CloudFormation stack. (But if you want to keep the recommendation stack separate from the main stack, you need to configure a way to pass information between the two stacks.) In addition, third-party tools, such as Terraform by HashiCorp, can also be used to provision Infrastructure as Code (IaC).

Using AWS Resilience Hub APIs to describe and manage application

As an alternative for describing and managing application using AWS Resilience Hub console, AWS Resilience Hub allows you to describe and manage applications using AWS Resilience Hub APIs. This chapter explains how to create an application using AWS Resilience Hub APIs. It also defines the sequence in which you need to execute the APIs and the parameter values you must provide with appropriate examples. For more information, see the following topics:

- the section called “Preparing the application”
- the section called “Running and analyzing the application”
- the section called “Modify your application”

Preparing the application

For preparing an application, you must first create an application, assign a resiliency policy, and then import the application resources from your input sources. For more information about the AWS Resilience Hub APIs that are used to prepare an application, see the following topics:

- the section called “Create an application”
- the section called “Create resiliency policy”
- the section called “Import application resource and monitor import status”
- the section called “Publish your application and assign resiliency policy”

Creating an application

To create a new application in AWS Resilience Hub, you must call the `CreateApp` API and provide a unique application name. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_CreateApp.html.

The following example shows how to create a new application newApp in AWS Resilience Hub using `CreateApp` API.

Request

```
aws resiliencehub create-app --name newApp
```

Response

```
{
    "app": {
        "appArn": "<App_ARN>",
        "name": "newApp",
        "creationTime": "2022-10-26T19:48:00.434000+03:00",
        "status": "Active",
        "complianceStatus": "NotAssessed",
        "resiliencyScore": 0.0,
        "tags": {},
        "assessmentSchedule": "Disabled"
    }
}
```

Creating resiliency policy

After creating the application, you must create a resiliency policy that enables you to understand your application's resiliency posture using `CreateResiliencyPolicy` API. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_CreateResiliencyPolicy.html.

The following example shows how to create `newPolicy` for your application in AWS Resilience Hub using `CreateResiliencyPolicy` API.

Request

```
aws resiliencehub create-resiliency-policy \
--policy-name newPolicy --tier NonCritical \
--policy '{"AZ": {"rtoInSecs": 172800,"rpoInSecs": 86400}, \
"Hardware": {"rtoInSecs": 172800,"rpoInSecs": 86400}, \
"Software": {"rtoInSecs": 172800,"rpoInSecs": 86400}}'
```

Response

```
{
    "policy": {
        "policyArn": "<Policy_ARN>",
        "policyName": "newPolicy",
        "policyDescription": "",
        "dataLocationConstraint": "AnyLocation",
        "tier": "NonCritical",
        "estimatedCostTier": "L1",
        "policy": {
            "AZ": {
                "rtoInSecs": 172800,
                "rpoInSecs": 86400
            },
            "Hardware": {
                "rtoInSecs": 172800,
                "rpoInSecs": 86400
            },
            "Software": {
                "rtoInSecs": 172800,
                "rpoInSecs": 86400
            }
        },
```

```
        "creationTime": "2022-10-26T20:48:05.946000+03:00",
        "tags": {}
    }
}
```

Importing resources from an input source and monitoring the import status

AWS Resilience Hub provides the following APIs to import resources to your application:

- `ImportResourcesToDraftAppVersion` – This API allows you to import resources to the draft version of your application from different input sources. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_ImportResourcesToDraftAppVersion.html.
- `PublishAppVersion` – This API publishes a new version of the application along with the updated AppComponents. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_PublishAppVersion.html.
- `DescribeDraftAppVersionResourcesImportStatus` – This API allows you to monitor the import status of your resources to an application version. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_DescribeDraftAppVersionResourcesImportStatus.html.

The following example shows how to import resources to your application in AWS Resilience Hub using `ImportResourcesToDraftAppVersion` API.

Request

```
aws resiliencehub import-resources-to-draft-app-version \
--app-arn <App_ARN> \
--terraform-sources '[{"s3StateFileUrl": <S3_URI>}]'
```

Response

```
{
    "appArn": "<App_ARN>",
    "appVersion": "draft",
    "sourceArns": [],
    "status": "Pending",
    "terraformSources": [
        {
```

```
            "s3StateFileUrl": <S3_URI>
        }
    ]
}
```

The following example shows how to manually add resources to your application in AWS Resilience Hub using `CreateAppVersionResource` API.

Request

```
aws resiliencehub create-app-version-resource \
--app-arn <App_ARN> \
--resource-name "backup-efs" \
--logical-resource-id '{"identifier": "backup-efs"}' \
--physical-resource-id '<Physical_resource_id_ARN>' \
--resource-type AWS::EFS::FileSystem \
--app-components '["new-app-component"]'
```

Response

```
{
    "appArn": "<App_ARN>",
    "appVersion": "draft",
    "physicalResource": {
        "resourceName": "backup-efs",
        "logicalResourceId": {
            "identifier": "backup-efs"
        },
        "physicalResourceId": {
            "identifier": "<Physical_resource_id_ARN>",
            "type": "Arn"
        },
        "resourceType": "AWS::EFS::FileSystem",
        "appComponents": [
            {
                "name": "new-app-component",
                "type": "AWS::ResilienceHub::StorageAppComponent",
                "id": "new-app-component"
            }
        ]
    }
}
```

The following example shows how to monitor the import status of your resources in AWS Resilience Hub using `DescribeDraftAppVersionResourcesImportStatus` API.

Request

```
aws resiliencehub describe-draft-app-version-resources-import-status \
--app-arn <App_ARN>
```

Response

```
{
    "appArn": "<App_ARN>",
    "appVersion": "draft",
    "status": "Success",
    "statusChangeTime": "2022-10-26T19:55:18.471000+03:00"
}
```

Publishing the draft version of your application and assigning a resiliency policy

Before running an assessment, you must first publish the draft version of your application and assign a resiliency policy to the released version of your application.

To publish the draft version of your application and assign a resiliency policy

1. To publish the draft version of your application, use `PublishAppVersion` API. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_PublishAppVersion.html.

   The following example shows how to publish the draft version of the application in AWS Resilience Hub using `PublishAppVersion` API.

   Request

   ```
   aws resiliencehub publish-app-version \
    --app-arn <App_ARN>
   ```

   Response

```
{
    "appArn": "<App_ARN>",
    "appVersion": "release"
}
```

2. Apply a resiliency policy to the released version of your application using UpdateApp API. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_UpdateApp.html.

   The following example shows how to apply a resiliency policy to the released version of an application in AWS Resilience Hub using UpdateApp API.

   Request

```
aws resiliencehub update-app \
--app-arn <App_ARN> \
--policy-arn <Policy_ARN>
```

   Response

```
{
    "app": {
        "appArn": "<App_ARN>",
        "name": "newApp",
        "policyArn": "<Policy_ARN>",
        "creationTime": "2022-10-26T19:48:00.434000+03:00",
        "status": "Active",
        "complianceStatus": "NotAssessed",
        "resiliencyScore": 0.0,
        "tags": {
            "resourceArn": "<App_ARN>"
        },
        "assessmentSchedule": "Disabled"
    }
}
```

Running and managing AWS Resilience Hub resiliency assessments

After you publish a new version of your application, you must run a new resiliency assessment and analyze the results to ensure that your application meets the estimated workload RTO and estimated RPO that are defined in your resiliency policy. The assessment compares each Application Component configuration to the policy and makes alarm, SOP, and test recommendations.

For more information, see the following topics:

- the section called “Run and monitor a resiliency assessment”
- the section called “Create resiliency policy”

Running and monitoring AWS Resilience Hub resiliency assessments

To run resiliency assessments in AWS Resilience Hub and monitor their status, you must use the following APIs:

- `StartAppAssessment` – This API creates a new assessment for an application. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_StartAppAssessment.html.
- `DescribeAppAssessment` – This API describes an assessment for the application and provides the completion status of the assessment. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_DescribeAppAssessment.html.

The following example shows how to start running a new assessment in AWS Resilience Hub using `StartAppAssessment` API.

Request

```
aws resiliencehub start-app-assessment \
--app-arn <App_ARN> \
--app-version release \
--assessment-name first-assessment
```

Response

```
{
    "assessment": {
```

```
        "appArn": "<App_ARN>",
        "appVersion": "release",
        "invoker": "User",
        "assessmentStatus": "Pending",
        "startTime": "2022-10-27T08:15:10.452000+03:00",
        "assessmentName": "first-assessment",
        "assessmentArn": "<Assessment_ARN>",
        "policy": {
            "policyArn": "<Policy_ARN>",
            "policyName": "newPolicy",
            "dataLocationConstraint": "AnyLocation",
            "policy": {
                "AZ": {
                    "rtoInSecs": 172800,
                    "rpoInSecs": 86400
                },
                "Hardware": {
                    "rtoInSecs": 172800,
                    "rpoInSecs": 86400
                },
                "Software": {
                    "rtoInSecs": 172800,
                    "rpoInSecs": 86400
                }
            }
        },
        "tags": {}
    }
}
```

The following example shows how to monitor the status of your assessment in AWS Resilience Hub using `DescribeAppAssessment` API. You can extract the status of your assessment from the `assessmentStatus` variable.

Request

```
aws resiliencehub describe-app-assessment \
--assessment-arn <Assessment_ARN>
```

Response

```
{
    "assessment": {
```

```
        "appArn": "<App_ARN>",
        "appVersion": "release",
        "cost": {
            "amount": 0.0,
            "currency": "USD",
            "frequency": "Monthly"
        },
        "resiliencyScore": {
            "score": 0.27,
            "disruptionScore": {
                "AZ": 0.42,
                "Hardware": 0.0,
                "Region": 0.0,
                "Software": 0.38
            }
        },
        "compliance": {
            "AZ": {
                "achievableRtoInSecs": 0,
                "currentRtoInSecs": 4500,
                "currentRpoInSecs": 86400,
                "complianceStatus": "PolicyMet",
                "achievableRpoInSecs": 0
            },
            "Hardware": {
                "achievableRtoInSecs": 0,
                "currentRtoInSecs": 2595601,
                "currentRpoInSecs": 2592001,
                "complianceStatus": "PolicyBreached",
                "achievableRpoInSecs": 0
            },
            "Software": {
                "achievableRtoInSecs": 0,
                "currentRtoInSecs": 4500,
                "currentRpoInSecs": 86400,
                "complianceStatus": "PolicyMet",
                "achievableRpoInSecs": 0
            }
        },
        "complianceStatus": "PolicyBreached",
        "assessmentStatus": "Success",
        "startTime": "2022-10-27T08:15:10.452000+03:00",
        "endTime": "2022-10-27T08:15:31.883000+03:00",
        "assessmentName": "first-assessment",
```

```
        "assessmentArn": "<Assessment_ARN>",
        "policy": {
            "policyArn": "<Policy_ARN>",
            "policyName": "newPolicy",
            "dataLocationConstraint": "AnyLocation",
            "policy": {
                "AZ": {
                    "rtoInSecs": 172800,
                    "rpoInSecs": 86400
                },
                "Hardware": {
                    "rtoInSecs": 172800,
                    "rpoInSecs": 86400
                },
                "Software": {
                    "rtoInSecs": 172800,
                    "rpoInSecs": 86400
                }
            }
        },
        "tags": {}
    }
}
```

Examining assessment results

After your assessment is completed successfully, you can examine the assessment results using the following APIs.

- `DescribeAppAssessment` – This API allows you to track the current status of your application against the resiliency policy. In addition, you can also extract the compliance status from `complianceStatus` variable, and the resiliency score for each disruption type from the `resiliencyScore` structure. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_DescribeAppAssessment.html.
- `ListAlarmRecommendations` – This API allows you to obtain the alarm recommendations using the Amazon Resource Name (ARN) of the assessment. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_ListAlarmRecommendations.html.

> **Note**
>
> To obtain the SOP and FIS test recommendations, use `ListSopRecommendations` and `ListTestRecommendations` APIs.

The following example shows how to obtain the alarm recommendations using the Amazon Resource Name (ARN) of the assessment using `ListAlarmRecommendations` API.

> **Note**
>
> To obtain the SOP and FIS test recommendations, replace with either `ListSopRecommendations` or `ListTestRecommendations`.

Request

```
aws resiliencehub list-alarm-recommendations \
--assessment-arn <Assessment_ARN>
```

Response

```
{
    "alarmRecommendations": [
        {
            "recommendationId": "78ece7f8-c776-499e-baa8-b35f5e8b8ba2",
            "referenceId": "app_common:alarm:synthetic_canary:2021-04-01",
            "name": "AWSResilienceHub-SyntheticCanaryInRegionAlarm_2021-04-01",
            "description": "A monitor for the entire application, configured to
 constantly verify that the application API/endpoints are available",
            "type": "Metric",
            "appComponentName": "appcommon",
            "items": [
                {
                    "resourceId": "us-west-2",
                    "targetAccountId": "12345678901",
                    "targetRegion": "us-west-2",
                    "alreadyImplemented": false
                }
            ],
```

```
            "prerequisite": "Make sure Amazon CloudWatch Synthetics is setup to monitor
 the application (see the <a href=\"https://docs.aws.amazon.com/AmazonCloudWatch/
latest/monitoring/CloudWatch_Synthetics_Canaries.html\" target=\"_blank\">docs</a>).
 \nMake sure that the Synthetics Name passed in the alarm dimension matches the name of
 the Synthetic Canary. It Defaults to the name of the application.\n"
        },
        {
            "recommendationId": "d9c72c58-8c00-43f0-ad5d-0c6e5332b84b",
            "referenceId": "efs:alarm:percent_io_limit:2020-04-01",
            "name": "AWSResilienceHub-EFSHighIoAlarm_2020-04-01",
            "description": "An alarm by AWS Resilience Hub that reports when Amazon EFS
 I/O load is more than 90% for too much time",
            "type": "Metric",
            "appComponentName": "storageappcomponent-rlb",
            "items": [
                {
                    "resourceId": "fs-0487f945c02f17b3e",
                    "targetAccountId": "12345678901",
                    "targetRegion": "us-west-2",
                    "alreadyImplemented": false
                }
            ]
        },
        {
            "recommendationId": "09f340cd-3427-4f66-8923-7f289d4a3216",
            "referenceId": "efs:alarm:mount_failure:2020-04-01",
            "name": "AWSResilienceHub-EFSMountFailureAlarm_2020-04-01",
            "description": "An alarm by AWS Resilience Hub that reports when volume
 failed to mount to EC2 instance",
            "type": "Metric",
            "appComponentName": "storageappcomponent-rlb",
            "items": [
                {
                    "resourceId": "fs-0487f945c02f17b3e",
                    "targetAccountId": "12345678901",
                    "targetRegion": "us-west-2",
                    "alreadyImplemented": false
                }
            ],
            "prerequisite": "* Make sure Amazon EFS utils are installed(see the <a
 href=\"https://github.com/aws/efs-utils#installation\" target=\"_blank\">docs</a>).
\n* Make sure cloudwatch logs are enabled in efs-utils (see the <a href=\"https://
github.com/aws/efs-utils#step-2-enable-cloudwatch-log-feature-in-efs-utils-config-
file-etcamazonefsefs-utilsconf\" target=\"_blank\">docs</a>).\n* Make sure that
```

```
you've configured `log_group_name` in `/etc/amazon/efs/efs-utils.conf`, for example:
`log_group_name = /aws/efs/utils`.\n* Use the created `log_group_name` in the
generated alarm. Find `LogGroupName: REPLACE_ME` in the alarm and make sure the
`log_group_name` is used instead of REPLACE_ME.\n"
      },
      {
          "recommendationId": "b0f57d2a-1220-4f40-a585-6dab1e79cee2",
          "referenceId": "efs:alarm:client_connections:2020-04-01",
          "name": "AWSResilienceHub-EFSHighClientConnectionsAlarm_2020-04-01",
          "description": "An alarm by AWS Resilience Hub that reports when client
connection number deviation is over the specified threshold",
          "type": "Metric",
          "appComponentName": "storageappcomponent-rlb",
          "items": [
              {
                  "resourceId": "fs-0487f945c02f17b3e",
                  "targetAccountId": "12345678901",
                  "targetRegion": "us-west-2",
                  "alreadyImplemented": false
              }
          ]
      },
      {
          "recommendationId": "15f49b10-9bac-4494-b376-705f8da252d7",
          "referenceId": "rds:alarm:health-storage:2020-04-01",
          "name": "AWSResilienceHub-RDSInstanceLowStorageAlarm_2020-04-01",
          "description": "Reports when database free storage is low",
          "type": "Metric",
          "appComponentName": "databaseappcomponent-hji",
          "items": [
              {
                  "resourceId": "terraform-20220623141426115800000001",
                  "targetAccountId": "12345678901",
                  "targetRegion": "us-west-2",
                  "alreadyImplemented": false
              }
          ]
      },
      {
          "recommendationId": "c1906101-cea8-4f77-be7b-60abb07621f5",
          "referenceId": "rds:alarm:health-connections:2020-04-01",
          "name": "AWSResilienceHub-RDSInstanceConnectionSpikeAlarm_2020-04-01",
          "description": "Reports when database connection count is anomalous",
          "type": "Metric",
```

```
            "appComponentName": "databaseappcomponent-hji",
            "items": [
                {
                    "resourceId": "terraform-20220623141426115800000001",
                    "targetAccountId": "12345678901",
                    "targetRegion": "us-west-2",
                    "alreadyImplemented": false
                }
            ]
        },
        {
            "recommendationId": "f169b8d4-45c1-4238-95d1-ecdd8d5153fe",
            "referenceId": "rds:alarm:health-cpu:2020-04-01",
            "name": "AWSResilienceHub-RDSInstanceOverUtilizedCpuAlarm_2020-04-01",
            "description": "Reports when database used CPU is high",
            "type": "Metric",
            "appComponentName": "databaseappcomponent-hji",
            "items": [
                {
                    "resourceId": "terraform-20220623141426115800000001",
                    "targetAccountId": "12345678901",
                    "targetRegion": "us-west-2",
                    "alreadyImplemented": false
                }
            ]
        },
        {
            "recommendationId": "69da8459-cbe4-4ba1-a476-80c7ebf096f0",
            "referenceId": "rds:alarm:health-memory:2020-04-01",
            "name": "AWSResilienceHub-RDSInstanceLowMemoryAlarm_2020-04-01",
            "description": "Reports when database free memory is low",
            "type": "Metric",
            "appComponentName": "databaseappcomponent-hji",
            "items": [
                {
                    "resourceId": "terraform-20220623141426115800000001",
                    "targetAccountId": "12345678901",
                    "targetRegion": "us-west-2",
                    "alreadyImplemented": false
                }
            ]
        },
        {
            "recommendationId": "67e7902a-f658-439e-916b-251a57b97c8a",
```

```
            "referenceId": "ecs:alarm:health-service_cpu_utilization:2020-04-01",
            "name": "AWSResilienceHub-ECSServiceHighCpuUtilizationAlarm_2020-04-01",
            "description": "An alarm by AWS Resilience Hub that triggers when CPU
utilization of ECS tasks of Service exceeds the threshold",
            "type": "Metric",
            "appComponentName": "computeappcomponent-nrz",
            "items": [
                {
                    "resourceId": "aws_ecs_service_terraform-us-east-1-demo",
                    "targetAccountId": "12345678901",
                    "targetRegion": "us-west-2",
                    "alreadyImplemented": false
                }
            ]
        },
        {
            "recommendationId": "fb30cb91-1f09-4abd-bd2e-9e8ee8550eb0",
            "referenceId": "ecs:alarm:health-service_memory_utilization:2020-04-01",
            "name": "AWSResilienceHub-ECSServiceHighMemoryUtilizationAlarm_2020-04-01",
            "description": "An alarm by AWS Resilience Hub for Amazon ECS that
indicates if the percentage of memory that is used in the service, is exceeding
specified threshold limit",
            "type": "Metric",
            "appComponentName": "computeappcomponent-nrz",
            "items": [
                {
                    "resourceId": "aws_ecs_service_terraform-us-east-1-demo",
                    "targetAccountId": "12345678901",
                    "targetRegion": "us-west-2",
                    "alreadyImplemented": false
                }
            ]
        },
        {
            "recommendationId": "1bd45a8e-dd58-4a8e-a628-bdbee234efed",
            "referenceId": "ecs:alarm:health-service_sample_count:2020-04-01",
            "name": "AWSResilienceHub-ECSServiceSampleCountAlarm_2020-04-01",
            "description": "An alarm by AWS Resilience Hub for Amazon ECS that triggers
if the count of tasks isn't equal Service Desired Count",
            "type": "Metric",
            "appComponentName": "computeappcomponent-nrz",
            "items": [
                {
                    "resourceId": "aws_ecs_service_terraform-us-east-1-demo",
```

```
                    "targetAccountId": "12345678901",
                    "targetRegion": "us-west-2",
                    "alreadyImplemented": false
                }
            ],
            "prerequisite": "Make sure the Container Insights on Amazon ECS is enabled:
 (see the <a href=\"https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
deploy-container-insights-ECS-cluster.html\" target=\"_blank\">docs</a>)."
        }
    ]
}
```

The following example shows how to obtain the configuration recommendations (recommendations on how to improve your current resiliency) using `ListAppComponentRecommendations` API.

Request

```
aws resiliencehub list-app-component-recommendations \
--assessment-arn <Assessment_ARN>
```

Response

```
{
    "componentRecommendations": [
        {
            "appComponentName": "computeappcomponent-nrz",
            "recommendationStatus": "MetCanImprove",
            "configRecommendations": [
                {
                    "cost": {
                        "amount": 0.0,
                        "currency": "USD",
                        "frequency": "Monthly"
                },
                    "appComponentName": "computeappcomponent-nrz",
                    "recommendationCompliance": {
                        "AZ": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": " Estimated time to restore
 cluster with volumes. (Estimate is based on averages, real time restore may vary).",
                            "expectedRpoInSecs": 86400,
```

```
                            "expectedRpoDescription": "Based on the frequency of the
backups"
                        },
                        "Hardware": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": " Estimated time to restore
cluster with volumes. (Estimate is based on averages, real time restore may vary).",
                            "expectedRpoInSecs": 86400,
                            "expectedRpoDescription": "Based on the frequency of the
backups"
                        },
                        "Software": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": " Estimated time to restore
cluster with volumes. (Estimate is based on averages, real time restore may vary).",
                            "expectedRpoInSecs": 86400,
                            "expectedRpoDescription": "Based on the frequency of the
backups"
                        }
                    },
                    "optimizationType": "LeastCost",
                    "description": "Current Configuration",
                    "suggestedChanges": [],
                    "haArchitecture": "BackupAndRestore",
                    "referenceId": "original"
                },
                {
                    "cost": {
                        "amount": 0.0,
                        "currency": "USD",
                        "frequency": "Monthly"
                    },
                    "appComponentName": "computeappcomponent-nrz",
                    "recommendationCompliance": {
                        "AZ": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": " Estimated time to restore
cluster with volumes. (Estimate is based on averages, real time restore may vary).",
                            "expectedRpoInSecs": 86400,
                            "expectedRpoDescription": "Based on the frequency of the
backups"
```

```
                        },
                        "Hardware": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": " Estimated time to restore
cluster with volumes. (Estimate is based on averages, real time restore may vary).",
                            "expectedRpoInSecs": 86400,
                            "expectedRpoDescription": "Based on the frequency of the
backups"
                        },
                        "Software": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": " Estimated time to restore
cluster with volumes. (Estimate is based on averages, real time restore may vary).",
                            "expectedRpoInSecs": 86400,
                            "expectedRpoDescription": "Based on the frequency of the
backups"
                        }
                    },
                    "optimizationType": "LeastChange",
                    "description": "Current Configuration",
                    "suggestedChanges": [],
                    "haArchitecture": "BackupAndRestore",
                    "referenceId": "original"
                },
                {
                    "cost": {
                        "amount": 14.74,
                        "currency": "USD",
                        "frequency": "Monthly"
                    },
                    "appComponentName": "computeappcomponent-nrz",
                    "recommendationCompliance": {
                        "AZ": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 0,
                            "expectedRtoDescription": "No expected downtime. You're
launching using EC2, with DesiredCount > 1 in multiple AZs and CapacityProviders with
MinSize > 1",
                            "expectedRpoInSecs": 0,
                            "expectedRpoDescription": "ECS Service state is saved on
Amazon EFS file system. No data loss is expected as objects are be stored in multiple
AZs."
```

```
                        },
                        "Hardware": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 0,
                            "expectedRtoDescription": "No expected downtime. You're
launching using EC2, with DesiredCount > 1 and CapacityProviders with MinSize > 1",
                            "expectedRpoInSecs": 0,
                            "expectedRpoDescription": "ECS Service state is saved on
Amazon EFS file system. No data loss is expected as objects are be stored in multiple
AZs."
                        },
                        "Software": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": " Estimated time to restore
cluster with volumes. (Estimate is based on averages, real time restore may vary).",
                            "expectedRpoInSecs": 86400,
                            "expectedRpoDescription": "Based on the frequency of the
backups"
                        }
                    },
                    "optimizationType": "BestAZRecovery",
                    "description": "Stateful Amazon ECS service with launch type Amazon
EC2 and Amazon EFS storage, deployed in multiple AZs. AWS Backup is used to backup
Amazon EFS and copy snapshots in-Region.",
                    "suggestedChanges": [
                        "Add AWS Auto Scaling Groups and Capacity Providers in multiple
AZs",
                        "Change desired count of the setup",
                        "Remove Amazon EBS volume"
                    ],
                    "haArchitecture": "BackupAndRestore",
                    "referenceId": "ecs:config:ec2-multi_az-efs-backups:2022-02-16"
                }
            ]
        },
        {
            "appComponentName": "databaseappcomponent-hji",
            "recommendationStatus": "MetCanImprove",
            "configRecommendations": [
                {
                    "cost": {
                        "amount": 0.0,
                        "currency": "USD",
```

```
                        "frequency": "Monthly"
                    },
                    "appComponentName": "databaseappcomponent-hji",
                    "recommendationCompliance": {
                        "AZ": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": "Estimated time to restore from
an RDS backup. (Estimates are averages based on size, real time may vary greatly from
estimate).",
                            "expectedRpoInSecs": 86400,
                            "expectedRpoDescription": "Estimate based on the backup
schedule. (Estimates are calculated from backup schedule, real time restore may
vary)."
                        },
                        "Hardware": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": "Estimated time to restore from
snapshot. (Estimates are averages based on size, real time may vary greatly from
estimate).",
                            "expectedRpoInSecs": 86400,
                            "expectedRpoDescription": "Estimate based on the backup
schedule. (Estimates are calculated from backup schedule, real time restore may
vary)."
                        },
                        "Software": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": "Estimated time to restore from
snapshot. (Estimates are averages based on size, real time may vary greatly from
estimate).",
                            "expectedRpoInSecs": 86400,
                            "expectedRpoDescription": "Estimate based on the backup
schedule. (Estimates are calculated from backup schedule, real time restore may
vary)."
                        }
                    },
                    "optimizationType": "LeastCost",
                    "description": "Current Configuration",
                    "suggestedChanges": [],
                    "haArchitecture": "BackupAndRestore",
                    "referenceId": "original"
                },
```

```
                {
                    "cost": {
                        "amount": 0.0,
                        "currency": "USD",
                        "frequency": "Monthly"
                    },
                    "appComponentName": "databaseappcomponent-hji",
                    "recommendationCompliance": {
                        "AZ": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": "Estimated time to restore from
an RDS backup. (Estimates are averages based on size, real time may vary greatly from
estimate).",
                            "expectedRpoInSecs": 86400,
                            "expectedRpoDescription": "Estimate based on the backup
schedule. (Estimates are calculated from backup schedule, real time restore may
vary)."
                        },
                        "Hardware": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": "Estimated time to restore from
snapshot. (Estimates are averages based on size, real time may vary greatly from
estimate).",
                            "expectedRpoInSecs": 86400,
                            "expectedRpoDescription": "Estimate based on the backup
schedule. (Estimates are calculated from backup schedule, real time restore may
vary)."
                        },
                        "Software": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 1800,
                            "expectedRtoDescription": "Estimated time to restore from
snapshot. (Estimates are averages based on size, real time may vary greatly from
estimate).",
                            "expectedRpoInSecs": 86400,
                            "expectedRpoDescription": "Estimate based on the backup
schedule. (Estimates are calculated from backup schedule, real time restore may
vary)."
                        }
                    },
                    "optimizationType": "LeastChange",
                    "description": "Current Configuration",
```

```
                    "suggestedChanges": [],
                    "haArchitecture": "BackupAndRestore",
                    "referenceId": "original"
                },
                {
                    "cost": {
                        "amount": 76.73,
                        "currency": "USD",
                        "frequency": "Monthly"
                    },
                    "appComponentName": "databaseappcomponent-hji",
                    "recommendationCompliance": {
                        "AZ": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 120,
                            "expectedRtoDescription": "Estimated time to promote a
secondary instance.",
                            "expectedRpoInSecs": 0,
                            "expectedRpoDescription": "Aurora data is automatically
replicated across multiple Availability Zones in a Region."
                        },
                        "Hardware": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 120,
                            "expectedRtoDescription": "Estimated time to promote a
secondary instance.",
                            "expectedRpoInSecs": 0,
                            "expectedRpoDescription": "Aurora data is automatically
replicated across multiple Availability Zones in a Region."
                        },
                        "Software": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 900,
                            "expectedRtoDescription": "Estimate time to backtrack to a
stable state.",
                            "expectedRpoInSecs": 300,
                            "expectedRpoDescription": "Estimate for latest restorable
time for point in time recovery."
                        }
                    },
                    "optimizationType": "BestAZRecovery",
                    "description": "Aurora database cluster with one read replica, with
backtracking window of 24 hours.",
                    "suggestedChanges": [
```

```
                        "Add read replica in the same Region",
                        "Change DB instance to a supported class (db.t3.small)",
                        "Change to Aurora",
                        "Enable cluster backtracking",
                        "Enable instance backup with retention period 7"
                    ],
                    "haArchitecture": "WarmStandby",
                    "referenceId": "rds:config:aurora-backtracking"
                }
            ]
        },
        {
            "appComponentName": "storageappcomponent-rlb",
            "recommendationStatus": "BreachedUnattainable",
            "configRecommendations": [
                {
                    "cost": {
                        "amount": 0.0,
                        "currency": "USD",
                        "frequency": "Monthly"
                    },
                    "appComponentName": "storageappcomponent-rlb",
                    "recommendationCompliance": {
                        "AZ": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 0,
                            "expectedRtoDescription": "No data loss in your system",
                            "expectedRpoInSecs": 0,
                            "expectedRpoDescription": "No data loss in your system"
                        },
                        "Hardware": {
                            "expectedComplianceStatus": "PolicyBreached",
                            "expectedRtoInSecs": 2592001,
                            "expectedRtoDescription": "No recovery option configured",
                            "expectedRpoInSecs": 2592001,
                            "expectedRpoDescription": "No recovery option configured"
                        },
                        "Software": {
                            "expectedComplianceStatus": "PolicyMet",
                            "expectedRtoInSecs": 900,
                            "expectedRtoDescription": "Time to recover Amazon EFS from
backup. (Estimate is based on averages, real time restore may vary).",
                            "expectedRpoInSecs": 86400,
```

```
                        "expectedRpoDescription": "Recovery Point Objective for
Amazon EFS from backups, derived from backup frequency"
                    }
                },
                "optimizationType": "BestAZRecovery",
                "description": "Amazon EFS with backups configured",
                "suggestedChanges": [
                    "Add additional availability zone"
                ],
                "haArchitecture": "MultiSite",
                "referenceId": "efs:config:with_backups:2020-04-01"
            },
            {
                "cost": {
                    "amount": 0.0,
                    "currency": "USD",
                    "frequency": "Monthly"
                },
                "appComponentName": "storageappcomponent-rlb",
                "recommendationCompliance": {
                    "AZ": {
                        "expectedComplianceStatus": "PolicyMet",
                        "expectedRtoInSecs": 0,
                        "expectedRtoDescription": "No data loss in your system",
                        "expectedRpoInSecs": 0,
                        "expectedRpoDescription": "No data loss in your system"
                    },
                    "Hardware": {
                        "expectedComplianceStatus": "PolicyBreached",
                        "expectedRtoInSecs": 2592001,
                        "expectedRtoDescription": "No recovery option configured",
                        "expectedRpoInSecs": 2592001,
                        "expectedRpoDescription": "No recovery option configured"
                    },
                    "Software": {
                        "expectedComplianceStatus": "PolicyMet",
                        "expectedRtoInSecs": 900,
                        "expectedRtoDescription": "Time to recover Amazon EFS from
backup. (Estimate is based on averages, real time restore may vary).",
                        "expectedRpoInSecs": 86400,
                        "expectedRpoDescription": "Recovery Point Objective for
Amazon EFS from backups, derived from backup frequency"
                    }
                },
```

```
                    "optimizationType": "BestAttainable",
                    "description": "Amazon EFS with backups configured",
                    "suggestedChanges": [
                        "Add additional availability zone"
                    ],
                    "haArchitecture": "MultiSite",
                    "referenceId": "efs:config:with_backups:2020-04-01"
                }
            ]
        }
    ]
}
```

Modifying your application

AWS Resilience Hub allows you to modify your application resources by editing a draft version of your application and publishing the changes to a new (published) version. AWS Resilience Hub uses the published version of your application, which includes the updated resources, for running resiliency assessments.

For more information, see the following topics:

- the section called “Manually add resources”
- the section called “Grouping resources into a single Application Component”
- the section called “Excluding a resource from an AppComponent”

Manually adding resources to your application

If the resource is not deployed as part of an input source, AWS Resilience Hub allows you to manually add the resource to your application using `CreateAppVersionResource` API. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_CreateAppVersionResource.html.

You must provide the following parameters to this API:

- Amazon Resource Name (ARN) of the application
- Logical ID of the resource
- Physical ID of the resource
- AWS CloudFormation type

The following example shows how to manually add resources to your application in AWS Resilience Hub using `CreateAppVersionResource` API.

Request

```
aws resiliencehub create-app-version-resource \
--app-arn <App_ARN> \
--resource-name "backup-efs" \
--logical-resource-id '{"identifier": "backup-efs"}' \
--physical-resource-id '<Physical_resource_id_ARN>' \
--resource-type AWS::EFS::FileSystem \
--app-components '["new-app-component"]'
```

Response

```
{
    "appArn": "<App_ARN>",
    "appVersion": "draft",
    "physicalResource": {
        "resourceName": "backup-efs",
        "logicalResourceId": {
            "identifier": "backup-efs"
        },
        "physicalResourceId": {
            "identifier": "<Physical_resource_id_ARN>",
            "type": "Arn"
        },
        "resourceType": "AWS::EFS::FileSystem",
        "appComponents": [
            {
                "name": "new-app-component",
                "type": "AWS::ResilienceHub::StorageAppComponent",
                "id": "new-app-component"
            }
        ]
    }
}
```

Grouping resources into a single Application Component

An Application Component (AppComponent) is a group of related AWS resources that work and fail as a single unit. For example, when you have cross-Region workloads that are used as

standby deployments. AWS Resilience Hub has rules governing which AWS resources can belong to which type of AppComponent. AWS Resilience Hub allows you to group resources into a single AppComponent using the following resource management APIs.

- `UpdateAppVersionResource` – This API updates the resource details of an application. For more information about this API, see UpdateAppVersionResource.
- `DeleteAppVersionAppComponent` – This API deletes the AppComponent from the application. For more information about this API, see DeleteAppVersionAppComponent.

The following example shows how to update the resource details of your application in AWS Resilience Hub using `DeleteAppVersionAppComponent` API.

Request

```
aws resiliencehub delete-app-version-app-component \
--app-arn <App_ARN> \
--id new-app-component
```

Response

```
{
    "appArn": "<App_ARN>",
    "appVersion": "draft",
    "appComponent": {
        "name": "new-app-component",
        "type": "AWS::ResilienceHub::StorageAppComponent",
        "id": "new-app-component"
    }
}
```

The following example shows how to delete the empty AppComponent that was created in the previous examples in AWS Resilience Hub using `UpdateAppVersionResource` API.

Request

```
aws resiliencehub delete-app-version-app-component \
--app-arn <App_ARN> \
--id new-app-component
```

Response

```
{
    "appArn": "<App_ARN>",
    "appVersion": "draft",
    "appComponent": {
        "name": "new-app-component",
        "type": "AWS::ResilienceHub::StorageAppComponent",
        "id": "new-app-component"
    }
}
```

Excluding a resource from an AppComponent

AWS Resilience Hub allows you to exclude resources from assessments using `UpdateAppVersionResource` API. These resources will not be considered while computing the resiliency of your application. For more information about this API, see https://docs.aws.amazon.com/resilience-hub/latest/APIReference/API_UpdateAppVersionResource.html.

> **Note**
>
> You can exclude only those resources that were imported from an input source.

The following example shows how to exclude a resource of your application in AWS Resilience Hub using `UpdateAppVersionResource` API.

Request

```
aws resiliencehub update-app-version-resource \
--app-arn <App_ARN> \
--resource-name "ec2instance-nvz" \
--excluded
```

Response

```
{
    "appArn": "<App_ARN>",
    "appVersion": "draft",
    "physicalResource": {
        "resourceName": "ec2instance-nvz",
```

```
        "logicalResourceId": {
            "identifier": "ec2",
            "terraformSourceName": "test.state.file"
        },
        "physicalResourceId": {
            "identifier": "i-0b58265a694e5ffc1",
            "type": "Native",
            "awsRegion": "us-west-2",
            "awsAccountId": "123456789101"
        },
        "resourceType": "AWS::EC2::Instance",
        "appComponents": [
            {
                "name": "computeappcomponent-nrz",
                "type": "AWS::ResilienceHub::ComputeAppComponent"
            }
        ]
    }
}
```

Security in AWS Resilience Hub

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to AWS Resilience Hub, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS Resilience Hub. The following topics show you how to configure AWS Resilience Hub to

meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS Resilience Hub resources.

Contents

- Data protection in AWS Resilience Hub
- Identity and Access Management for AWS Resilience Hub
- Infrastructure security in AWS Resilience Hub

Data protection in AWS Resilience Hub

The AWS shared responsibility model applies to data protection in AWS Resilience Hub. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see Data Privacy FAQ. For information about data protection in Europe, see the General Data Protection Regulation (GDPR) Center.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Resilience Hub or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Encryption at rest

AWS Resilience Hub encrypts your data at rest. Data in AWS Resilience Hub is encrypted at rest using transparent server-side encryption. This helps reduce the operational burden and complexity involved in protecting sensitive data. With encryption at rest, you can build security-sensitive applications that meet encryption compliance and regulatory requirements.

Encryption in transit

AWS Resilience Hub encrypts data in transit between the service and other integrated AWS services. All data that passes between AWS Resilience Hub and integrated services is encrypted using Transport Layer Security (TLS). AWS Resilience Hub provides preconfigured actions for specific types of targets across AWS services, and supports actions for target resources.

Identity and Access Management for AWS Resilience Hub

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use AWS Resilience Hub resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Audience
- Authenticating with identities
- Managing access using policies
- How AWS Resilience Hub works with IAM
- Setup IAM roles and permissions
- Troubleshooting AWS Resilience Hub identity and access

- AWS Resilience Hub access permissions reference
- AWS managed policies for AWS Resilience Hub
- AWS Resilience Hub personas and IAM permissions reference
- Importing Terraform state file into AWS Resilience Hub
- Enabling AWS Resilience Hub access to your Amazon Elastic Kubernetes Service cluster
- Enabling AWS Resilience Hub to publish to your Amazon Simple Notification Service topics
- Limiting permissions to include or exclude AWS Resilience Hub recommendations

Audience

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** - request permissions from your administrator if you cannot access features (see Troubleshooting AWS Resilience Hub identity and access)
- **Service administrator** - determine user access and submit permission requests (see How AWS Resilience Hub works with IAM)
- **IAM administrator** - write policies to manage access (see Identity-based policy examples for AWS Resilience Hub)

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you

don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users to use federation with an identity provider to access AWS services using temporary credentials.

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.

- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS Resilience Hub works with IAM

Before you use IAM to manage access to AWS Resilience Hub, learn what IAM features are available to use with AWS Resilience Hub.

IAM features you can use with AWS Resilience Hub

IAM feature	AWS Resilience Hub support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	No
ABAC (tags in policies)	Partial
Temporary credentials	Yes
Forward access sessions (FAS)	Yes
Service roles	Yes

To get a high-level view of how AWS Resilience Hub and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for AWS Resilience Hub

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for AWS Resilience Hub

To view examples of AWS Resilience Hub identity-based policies, see Identity-based policy examples for AWS Resilience Hub.

Resource-based policies within AWS Resilience Hub

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for AWS Resilience Hub

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

To see a list of AWS Resilience Hub actions, see Actions defined by AWS Resilience Hub in the *Service Authorization Reference*.

Policy actions in AWS Resilience Hub use the following prefix before the action:

```
resiliencehub
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "resiliencehub:action1",
      "resiliencehub:action2"
         ]
```

To view examples of AWS Resilience Hub identity-based policies, see Identity-based policy examples for AWS Resilience Hub.

Policy resources for AWS Resilience Hub

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of AWS Resilience Hub resource types and their ARNs, see Resources defined by AWS Resilience Hub in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by AWS Resilience Hub.

To view examples of AWS Resilience Hub identity-based policies, see Identity-based policy examples for AWS Resilience Hub.

Policy condition keys for AWS Resilience Hub

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element specifies when statements execute based on defined criteria. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of AWS Resilience Hub condition keys, see Condition keys for AWS Resilience Hub in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by AWS Resilience Hub.

To view examples of AWS Resilience Hub identity-based policies, see Identity-based policy examples for AWS Resilience Hub.

ACLs in AWS Resilience Hub

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with AWS Resilience Hub

Supports ABAC (tags in policies): Partial

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with AWS Resilience Hub

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM and AWS services that work with IAM in the *IAM User Guide*.

Forward access sessions for AWS Resilience Hub

Supports forward access sessions (FAS): Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see Forward access sessions.

Service roles for AWS Resilience Hub

Supports service roles: Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
>
> Changing the permissions for a service role might break AWS Resilience Hub functionality. Edit service roles only when AWS Resilience Hub provides guidance to do so.

Identity-based policy examples for AWS Resilience Hub

By default, users and roles don't have permission to create or modify AWS Resilience Hub resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by AWS Resilience Hub, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS Resilience Hub in the *Service Authorization Reference*.

Topics

- Policy best practices
- Using the AWS Resilience Hub console
- Allow users to view their own permissions
- Listing available AWS Resilience Hub applications
- Starting an application assessment
- Deleting an application assessment
- Creating a recommendation template for a specific application
- Deleting a recommendation template for a specific application
- Updating an application with a specific resiliency policy

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS Resilience Hub resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.

- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the AWS Resilience Hub console

To access the AWS Resilience Hub console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the AWS Resilience Hub resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the AWS Resilience Hub console, also attach the AWS Resilience Hub *ConsoleAccess* or *ReadOnly* AWS managed policy to the entities. For more information, see Adding permissions to a user in the *IAM User Guide*.

The following policy grants users the permission to list and view all resources in the AWS Resilience Hub console, but not to create, update, or delete them.

JSON

```
{
     "Version":"2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                 "resiliencehub:List*",
                 "resiliencehub:Describe*"
             ],
             "Resource": "*"
         }
     ]
 }
```

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
```

```
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Listing available AWS Resilience Hub applications

The following policy grants users the permission to list the available AWS Resilience Hub applications.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "resiliencehub:ListApps"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
```

```
}
```

Starting an application assessment

The following policy grants users the permission to start an assessment for a specific AWS Resilience Hub application.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "resiliencehub:StartAppAssessment"
      ],
      "Resource": [
        "arn:aws:resiliencehub:*:*:app/appId"
      ]
    }
  ]
}
```

Deleting an application assessment

The following policy grants users the permission to delete an assessment for a specific AWS Resilience Hub application.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
```

```
            "resiliencehub:DeleteAppAssessment"
        ],
        "Resource": [
          "arn:aws:resiliencehub:*:*:app/appId"
        ]
      }
    ]
}
```

Creating a recommendation template for a specific application

The following policy grants users the permission to create a recommendation template for a specific AWS Resilience Hub application.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "resiliencehub:CreateRecommendationTemplate"
      ],
      "Resource": [
        "arn:aws:resiliencehub:*:*:app/appId"
      ]
    }
  ]
}
```

Deleting a recommendation template for a specific application

The following policy grants users the permission to delete a recommendation template for a specific AWS Resilience Hub application.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "resiliencehub:DeleteRecommendationTemplate"
      ],
      "Resource": [
        "arn:aws:resiliencehub:*:*:app/appId"
      ]
    }
  ]
}
```

Updating an application with a specific resiliency policy

The following policy grants users the permission to update an AWS Resilience Hub application with a specific resiliency policy.

Setup IAM roles and permissions

AWS Resilience Hub allows you to configure the IAM roles you would like to use while running assessments for your application. There are multiple ways to configure AWS Resilience Hub to gain read-only access to your application resources. However, AWS Resilience Hub recommends the following ways:

- **Role based access** – This role is defined and used in the current account. AWS Resilience Hub will assume this role to access the resources of your application.

  To provide role-based access, the role must include the following:

  - Read-only permission to read your resources (AWS Resilience Hub recommends you to use the `AWSResilienceHubAsssessmentExecutionPolicy` managed policy).
  - Trust policy to assume this role, which allows AWS Resilience Hub Service Principal to assume this role. If you don't have such a role configured in your account, AWS Resilience Hub will

display the instructions to create that role. For more information, see the section called "Setup permissions".

> **Note**
>
> If you provide only the invoker role name and if your resources are located in another account, AWS Resilience Hub will use this role name in the other accounts to access the cross-account resources. Optionally, you can configure the role ARNs for other accounts, which will be used instead of the invoker role name.

- **Current IAM user access** – AWS Resilience Hub will use the current IAM user to access your application resources. When your resources are in a different account, AWS Resilience Hub will assume the following IAM roles to access the resources:
  - `AwsResilienceHubAdminAccountRole` in the current account
  - `AwsResilienceHubExecutorAccountRole` in other accounts

  In addition, when you configure a scheduled assessment, AWS Resilience Hub will assume the `AwsResilienceHubPeriodicAssessmentRole` role. However, using `AwsResilienceHubPeriodicAssessmentRole` is not advised because you must manually configure roles and permissions, and some functionalities (such as **Drift notification**) might not work as expected.

Troubleshooting AWS Resilience Hub identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with AWS Resilience Hub and IAM.

Topics

- I am not authorized to perform an action in AWS Resilience Hub
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my AWS Resilience Hub resources

I am not authorized to perform an action in AWS Resilience Hub

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `resiliencehub:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 resiliencehub:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `resiliencehub:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to AWS Resilience Hub.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in AWS Resilience Hub. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my AWS Resilience Hub resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support

resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether AWS Resilience Hub supports these features, see How AWS Resilience Hub works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

AWS Resilience Hub access permissions reference

You can use AWS Identity and Access Management (IAM) to manage access to the application resources and create IAM policies that apply to users, groups, or roles.

Every AWS Resilience Hub application can be configured to use the the section called “Invoker role” (an IAM role), or use the current IAM user permissions (along with a set of predefined roles for cross-account and scheduled assessment). In this role, you can attach a policy that defines the permissions required by AWS Resilience Hub to access other AWS resources or application resources. The invoker role must have a trust policy that is added to AWS Resilience Hub Service Principal.

To manage permissions for your application, we recommend using the section called “AWS managed policies”. You can use these managed policies without any modifications, or you can use them as a starting point to write your own restrictive policies. Policies can restrict user permissions at the resource level for different actions by using additional optional conditions.

If your application resources are in different accounts (secondary/resource accounts), you must setup a new role in each account that contains your application resources.

> **Note**
>
> If you define VPC endpoints for your workload resources, ensure that the VPC endpoint policies provide read-only access to AWS Resilience Hub for accessing the resources. For more information, see Control access to VPC endpoints using endpoint policies.

Topics

- the section called “Using IAM role”
- the section called “Using current IAM user permissions”

Using IAM role

AWS Resilience Hub will use a predefined existing IAM role to access your resources in the primary account or secondary/resources account. This is the recommended permission option to access your resources.

Topics

- the section called “Invoker role”
- the section called “Roles in different AWS account for cross-account access”

Invoker role

The AWS Resilience Hub invoker role is an AWS Identity and Access Management (IAM) role that AWS Resilience Hub assumes to access AWS services and resources. For example, you might create an invoker role that has permission to access your CFN template and the resource it creates. This page provides information on how to create, view, and manage an application invoker role.

When you create an application, you provide an invoker role. AWS Resilience Hub assumes this role to access your resources when you import resources or start an assessment. For AWS Resilience Hub to properly assume your invoker role, the role's trust policy must specify the AWS Resilience Hub service principal (**resiliencehub.amazonaws.com**) as a trusted service.

To view the application's invoker role, choose **Applications** from the navigation pane, and then choose **Update permissions** from **Actions** menu in the **Application** page.

You can add or remove permissions from an application invoker role at any time, or configure your application to use a different role for accessing application resources.

Topics

- the section called “Creating an invoker role in the IAM console”
- the section called “Managing roles with the IAM API”
- the section called “Defining trust policy using JSON file”

Creating an invoker role in the IAM console

To enable AWS Resilience Hub to access AWS services and resources, you must create an invoker role in the primary account using the IAM console. For more information about creating roles using IAM console, see Creating a role for an AWS service (console).

To create an invoker role in the primary account using IAM console

1. Open the IAM console at `https://console.aws.amazon.com/iam/`.
2. From the navigation pane, choose **Roles** and then choose **Create role**.
3. Select **Custom Trust Policy**, copy the following policy in the **Custom trust policy** window, and then choose **Next**.

> **Note**
>
> If your resources are in different accounts, you have to create a role in each of those accounts, and use the secondary account trust policy for the other accounts.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "resiliencehub.amazonaws.com"
      },
```

```
      "Action": "sts:AssumeRole"
    }
  ]
}
```

4. In the **Permissions policies** section of **Add permissions** page, enter `AWSResilienceHubAsssessmentExecutionPolicy` in the **Filter policies by property or policy name and press enter** box.
5. Select the policy and choose **Next**.
6. In **Role details** section, enter a unique role name (such as `AWSResilienceHubAssessmentRole`) in the **Role name** box.

   This field accepts only alphanumeric and '`+=,.@-_/`' characters.
7. (Optional) Enter a description about the role in the **Description** box.
8. Choose **Create Role**.

   To edit the use cases and permissions, in step 6, choose **Edit** button that is located to the right of **Step 1: Select trusted entities** or **Step 2: Add permissions** sections.

After creating the invoker role and the resource role (if applicable), you can configure your application to use these roles.

> **Note**
>
> You must have an `iam:passRole` permission in your current IAM user/role to the invoker role when creating or updating the application. However, you do not need this permission to run an assessment.

Managing roles with the IAM API

A role's trust policy gives the specified principal's permission to assume the role. To create the roles using AWS Command Line Interface (AWS CLI), use the `create-role` command. While using this command, you can specify the trust policy inline. The following example shows how to grant the AWS Resilience Hub service the principal permission to assume your role.

> **Note**
>
> The requirement to escape quotes (' ') in the JSON string may vary based on your shell version.

Sample `create-role`

```
aws iam create-role --role-name AWSResilienceHubAssessmentRole --assume-role-policy-
document '{
  "Version": "2012-10-17",       "Statement":
  [
    {
      "Effect": "Allow",
      "Principal": {"Service": "resiliencehub.amazonaws.com"},
      "Action": "sts:AssumeRole"
    }
  ]
}'
```

Defining trust policy using JSON file

You can define the trust policy for the role using a separate JSON file and then run the `create-role` command. In the following example, **`trust-policy.json`** is a file that contains the trust policy in the current directory. This policy is attached to a role by running **`create-role`** command. The output of the **`create-role`** command is shown in the **Sample Output**. To add permissions to the role, use the **attach-policy-to-role** command and you can start by adding the `AWSResilienceHubAsssessmentExecutionPolicy` managed policy. For more information about this managed policy, see the section called “AWSResilienceHubAsssessmentExecutionPolicy”.

Sample `trust-policy.json`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": "resiliencehub.amazonaws.com"
```

```
        },
        "Action": "sts:AssumeRole"
    }]
}
```

Sample `create-role`

```
aws iam create-role --role-name AWSResilienceHubAssessmentRole --assume-role-policy-document file://trust-policy.json
```

Sample Output

Sample `attach-policy-to-role`

```
aws iam attach-role-policy --role-name AWSResilienceHubAssessmentRole --policy-arn arn:aws:iam::aws:policy/AWSResilienceHubAsssessmentExecutionPolicy
```

Roles in different AWS account for cross-account access - optional

When your resources are located in secondary/resource accounts, you must create roles in each of these accounts to enable AWS Resilience Hub to successfully assess your application. The role creation procedure is similar to the invoker role creation process, except for the trust policy configuration.

> **Note**
>
> You must create the roles in secondary accounts where the resources are located.

Topics

- the section called “Creating a role in the IAM console for secondary/resource accounts”
- the section called “Managing roles with the IAM API”
- the section called “Defining trust policy using JSON file”

Creating a role in the IAM console for secondary/resource accounts

To enable AWS Resilience Hub to access AWS services and resources in other AWS accounts, you must create roles in each of these accounts.

To create a role in the IAM console for the secondary/resource accounts using IAM console

1. Open the IAM console at `https://console.aws.amazon.com/iam/`.
2. From the navigation pane, choose **Roles** and then choose **Create role**.
3. Select **Custom Trust Policy**, copy the following policy in the **Custom trust policy** window, and then choose **Next**.

   > **Note**
   >
   > If your resources are in different accounts, you have to create a role in each of those accounts and use the secondary account trust policy for the other accounts.

4. In the **Permissions policies** section of **Add permissions** page, enter `AWSResilienceHubAsssessmentExecutionPolicy` in the **Filter policies by property or policy name and press enter** box.
5. Select the policy and choose **Next**.
6. In **Role details** section, enter a unique role name (such as `AWSResilienceHubAssessmentRole`) in the **Role name** box.
7. (Optional) Enter a description about the role in the **Description** box.
8. Choose **Create Role**.

   To edit the use cases and permissions, in step 6, choose **Edit** button that is located to the right of **Step 1: Select trusted entities** or **Step 2: Add permissions** sections.

In addition, you also need to add the `sts:assumeRole` permission to the invoker role to enable it to assume the roles in your secondary accounts.

Add the following policy to your invoker role for each of the secondary roles you created:

```
{
    "Effect": "Allow",
    "Resource": [
      "arn:aws:iam::secondary_account_id_1:role/RoleInSecondaryAccount_1",
      "arn:aws:iam::secondary_account_id_2:role/RoleInSecondaryAccount_2",
      ...
      ],
      "Action": [
        "sts:AssumeRole"
```

```
      ]
}
```

Managing roles with the IAM API

A role's trust policy gives the specified principal's permission to assume the role. To create the roles using AWS Command Line Interface (AWS CLI), use the `create-role` command. When using this command, you can specify the trust policy inline. The following example shows how to grant the AWS Resilience Hub service principal permission to assume your role.

> **Note**
>
> The requirement to escape quotes (`\"`) in the JSON string may vary based on your shell version.

Sample `create-role`

```
aws iam create-role --role-name AWSResilienceHubAssessmentRole --assume-role-policy-
document '{"Version": "2012-10-17",       "Statement": [{"Effect": "Allow","Principal":
 {"AWS": ["arn:aws:iam::primary_account_id:role/InvokerRoleName"]},"Action":
 "sts:AssumeRole"}]}'
```

You can also define the trust policy for the role using a separate JSON file. In the following example, `trust-policy.json` is a file in the current directory.

Defining trust policy using JSON file

You can define the trust policy for the role using a separate JSON file and then run the `create-role` command. In the following example, **`trust-policy.json`** is a file that contains the trust policy in the current directory. This policy is attached to a role by running **`create-role`** command. The output of the **`create-role`** command is shown in the **Sample Output**. To add permissions to a role, use the **attach-policy-to-role** command and you can start by adding the `AWSResilienceHubAsssessmentExecutionPolicy` managed policy. For more information about this managed policy, see the section called “AWSResilienceHubAsssessmentExecutionPolicy”.

Sample `trust-policy.json`

Sample `create-role`

```
aws iam create-role --role-name AWSResilienceHubAssessmentRole --assume-role-policy-
document file://trust-policy.json
```

Sample Output

Sample `attach-policy-to-role`

`aws iam attach-role-policy --role-name AWSResilienceHubAssessmentRole --policy-arn arn:aws:iam::aws:policy/AWSResilienceHubAsssessmentExecutionPolicy`.

Using current IAM user permissions

Use this method if you want to use your current IAM user permissions to create and run an assessment. You can attach the `AWSResilienceHubAsssessmentExecutionPolicy` managed policy to your IAM user or a Role associated with your user.

Single account setup

Using the managed policy mentioned above is enough to run an assessment on an application which is managed in the same account as the IAM user.

Scheduled assessment setup

You must create a new role `AwsResilienceHubPeriodicAssessmentRole` to enable AWS Resilience Hub to perform scheduled assessment related tasks.

> **Note**
>
> - While using the role-based access (with the invoker role mentioned above) this step is not required.
> - The role name must be `AwsResilienceHubPeriodicAssessmentRole`.

To enable AWS Resilience Hub to perform scheduled assessment related tasks

1. Attach the `AWSResilienceHubAsssessmentExecutionPolicy` managed policy to the role.
2. Add the following policy, where `primary_account_id` is the AWS account where the application is defined and will run the assessment. In addition,

you must add the associated trust policy for the scheduled assessment's role, (`AwsResilienceHubPeriodicAssessmentRole`), which gives permissions for the AWS Resilience Hub service to assume the scheduled assessment's role.

Trust policy for the scheduled assessment's role (`AwsResilienceHubPeriodicAssessmentRole`)

JSON

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

Cross-account setup

The following IAM permissions policies are required if you're using AWS Resilience Hub with multiple accounts. Each AWS account might need different permissions depending on your use case. While setting up AWS Resilience Hub for cross-account access, the following accounts and roles are considered:

- **Primary account** – AWS account in which you want to create the application and run assessments.
- **Secondary/Resource account(s)** – AWS account(s) where the resources are located.

> **Note**
>
> - While using the role-based access (with the invoker role mentioned above) this step is not required.

- For more information about configuring permissions to access Amazon Elastic Kubernetes Service, see the section called "Enabling AWS Resilience Hub access to your Amazon EKS cluster".

Primary account setup

You must create a new role `AwsResilienceHubAdminAccountRole` in the primary account and enable AWS Resilience Hub access to assume it. This role will be used to access another role in your AWS account that contains your resources. It should not have permissions to read resources.

Note

- The role name must be `AwsResilienceHubAdminAccountRole`.
- It must be created in the primary account.
- Your current IAM user/role must have the `iam:assumeRole` permission to assume this role.
- Replace `secondary_account_id_1/2/...` with the relevant secondary account identifiers.

The following policy provides executor permissions to your role for accessing resources in another role in your AWS account:

The trust policy for the admin role (`AwsResilienceHubAdminAccountRole`) is as follows:

Secondary/Resource account(s) setup

In each of your secondary accounts, you must create a new `AwsResilienceHubExecutorAccountRole` and enable the admin role created above to assume this role. Since this role will be used by AWS Resilience Hub to scan and assess your application resources, it will also require the appropriate permissions.

However, you must attach the `AWSResilienceHubAsssessmentExecutionPolicy` managed policy to the role and attach the executor role policy.

The executor role trust policy is as follows:

AWS managed policies for AWS Resilience Hub

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWSResilienceHubAsssessmentExecutionPolicy

You can attach the `AWSResilienceHubAsssessmentExecutionPolicy` to your IAM identities. While running an assessment, this policy grants access permissions to other AWS services for executing assessments.

Permission details

This policy provides adequate permissions to publish alarms, AWS FIS and SOP templates to your Amazon Simple Storage Service (Amazon S3) bucket. The Amazon S3 bucket name must start with `aws-resilience-hub-artifacts-`. If you wish to publish to another Amazon S3 bucket, you can do that while calling `CreateRecommendationTemplate` API. For more information, see CreateRecommendationTemplate.

This policy includes the following permissions:

- Amazon CloudWatch (CloudWatch) – Gets all the implemented alarms that you set up in Amazon CloudWatch to monitor the application. In addition, we use `cloudwatch:PutMetricData` to

publish CloudWatch metrics for the resiliency score of the application in the `ResilienceHub` namespace.

- Amazon Data Lifecycle Manager – Gets and provides `Describe` permissions for Amazon Data Lifecycle Manager resources that are associated with your AWS account.
- Amazon DevOps Guru – Lists and provides `Describe` permissions for Amazon DevOps Guru resources that are associated with your AWS account.
- Amazon DocumentDB – Lists and provides `Describe` permissions for Amazon DocumentDB resources that are associated with your AWS account.
- Amazon DynamoDB (DynamoDB) – Lists and provides `Describe` permissions for Amazon DynamoDB resources that are associated with your AWS account.
- Amazon ElastiCache (ElastiCache) – Provides `Describe` permissions for ElastiCache resources that are associated with your AWS account.
- Amazon ElastiCache (Redis OSS) Serverless (ElastiCache (Redis OSS) Serverless) – Provides `Describe` permissions for ElastiCache (Redis OSS) Serverless configurations that are associated with your AWS account.
- Amazon Elastic Compute Cloud (Amazon EC2) – Lists and provides `Describe` permissions for Amazon EC2 resources that are associated with your AWS account.
- Amazon Elastic Container Registry (Amazon ECR) – Provides `Describe` permissions for Amazon ECR resources that are associated with your AWS account.
- Amazon Elastic Container Service (Amazon ECS) – Provides `Describe` permissions for Amazon ECS resources that are associated with your AWS account.
- Amazon Elastic File System (Amazon EFS) – Provides `Describe` permissions for Amazon EFS resources that are associated with your AWS account.
- Amazon Elastic Kubernetes Service (Amazon EKS) – Lists and provides `Describe` permissions for Amazon EKS resources that are associated with your AWS account.
- Amazon EC2 Auto Scaling – Lists and provides `Describe` permissions for Amazon EC2 Auto Scaling resources that are associated with your AWS account.
- Amazon EC2 Systems Manager (SSM) – Provides `Describe` permissions for SSM resources that are associated with your AWS account.
- AWS Fault Injection Service (AWS FIS) – Lists and provides `Describe` permissions for AWS FIS experiments and experiment templates that are associated with your AWS account.
- Amazon FSx for Windows File Server (Amazon FSx) – Lists and provides `Describe` permissions for Amazon FSx resources that are associated with your AWS account.

- Amazon RDS – Lists and provides `Describe` permissions for Amazon RDS resources that are associated with your AWS account.
- Amazon Route 53 (Route 53) – Lists and provides `Describe` permissions for Route 53 resources that are associated with your AWS account.
- Amazon Route 53 Resolver – Lists and provides `Describe` permissions for Amazon Route 53 Resolver resources that are associated with your AWS account.
- Amazon Simple Notification Service (Amazon SNS) – Lists and provides `Describe` permissions for Amazon SNS resources that are associated with your AWS account.
- Amazon Simple Queue Service (Amazon SQS) – Lists and provides `Describe` permissions for Amazon SQS resources that are associated with your AWS account.
- Amazon Simple Storage Service (Amazon S3) – Lists and provides `Describe` permissions for Amazon S3 resources that are associated with your AWS account.

> **Note**
>
> While running an assessment, if there are any missing permissions that needs to be updated from Managed policies, AWS Resilience Hub will successfully complete the assessment using s3:GetBucketLogging permission. However, AWS Resilience Hub will display a warning message that lists the missing permissions and will provide a grace period to add the same. If you do not add the missing permissions within the specified grace period, the assessment will fail.

- AWS Backup – Lists and gets `Describe` permissions for Amazon EC2 Auto Scaling resources that are associated with your AWS account.
- AWS CloudFormation – Lists and gets `Describe` permissions for resources on AWS CloudFormation stacks that are associated with your AWS account.
- AWS DataSync – Lists and provides `Describe` permissions for AWS DataSync resources that are associated with your AWS account.
- Directory Service – Lists and provides `Describe` permissions for Directory Service resources that are associated with your AWS account.
- AWS Elastic Disaster Recovery (Elastic Disaster Recovery) – Provides `Describe` permissions for Elastic Disaster Recovery resources that are associated with your AWS account.
- AWS Lambda (Lambda) – Lists and provides `Describe` permissions for Lambda resources that are associated with your AWS account.

- AWS Resource Groups (Resource Groups) – Lists and provides `Describe` permissions for Resource Groups resources that are associated with your AWS account.
- AWS Service Catalog (Service Catalog) – Lists and provides `Describe` permissions for Service Catalog resources that are associated with your AWS account.
- AWS Step Functions – Lists and provides `Describe` permissions for AWS Step Functions resources that are associated with your AWS account.
- Elastic Load Balancing – Lists and provides `Describe` permissions for Elastic Load Balancing resources that are associated with your AWS account.
- `ssm:GetParametersByPath` – We use this permission to manage CloudWatch alarms, tests, or SOPs that are configured for your application.

The following IAM policy is required for an AWS account to add permissions for users, user-groups, and roles that provide required permissions for your team to access AWS services while running assessments.

AWS Resilience Hub updates to AWS managed policies

View details about updates to AWS managed policies for AWS Resilience Hub since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the AWS Resilience Hub Document history page.

Change	Description	Date
AWSResilienceHubAssessmentExecutionPolicy – Change	AWS Resilience Hub updated the `AWSResilienceHubAssessmentExecutionPolicy` to grant `List` and `Get` permissions to allow you to access experiments from AWS FIS while running assessments.	December 17, 2024
AWSResilienceHubAssessmentExecutionPolicy – Change	AWS Resilience Hub updated the `AWSResilienceHubAssessmentExecution`	September 25, 2024

Change	Description	Date
	`Policy` to grant `Describe` permissions to allow you to access resources and configurations on Amazon ElastiCache (Redis OSS) Serverless while running assessments.	
AWSResilienceHubAssessmentExecutionPolicy – Change	AWS Resilience Hub updated the `AWSResilienceHubAssessmentExecutionPolicy` to grant `Describe` permissions to allow you to access resources and configurations on Amazon DocumentDB, Elastic Load Balancing, and AWS Lambda while running assessments.	August 01, 2024
AWSResilienceHubAssessmentExecutionPolicy – Change	AWS Resilience Hub updated the `AWSResilienceHubAssessmentExecutionPolicy` to grant `Describe` permissions to allow you to read the Amazon FSx for Windows File Server configuration while running assessments.	March 26, 2024

Change	Description	Date
AWSResilienceHubAssessmentExecutionPolicy – Change	AWS Resilience Hub updated the `AWSResilienceHubAssessmentExecutionPolicy` to grant `Describe` permissions to allow you to read the AWS Step Functions configuration while running assessments.	October 30, 2023
AWSResilienceHubAssessmentExecutionPolicy – Change	AWS Resilience Hub updated the `AWSResilienceHubAssessmentExecutionPolicy` to grant `Describe` permissions to allow you to access resources on Amazon RDS while running assessments.	October 5, 2023
AWSResilienceHubAssessmentExecutionPolicy – New	This AWS Resilience Hub policy provides access to other AWS services for running assessments.	June 26, 2023
AWS Resilience Hub started tracking changes	AWS Resilience Hub started tracking changes for its AWS managed policies.	June 15, 2023

AWS Resilience Hub personas and IAM permissions reference

You can grant the IAM permissions to personas that are required to work with AWS Resilience Hub by using `AWSResilienceHubAsssessmentExecutionPolicy` AWS managed policy and one of the following persona-specific policies. For more information about AWS managed policy, see the section called "AWSResilienceHubAsssessmentExecutionPolicy".

Policies for personas suggested by AWS Resilience Hub:

- IAM permissions for Infrastructure application manager persona
- IAM permissions for Business continuity manager persona
- IAM permissions for Application owner persona
- IAM permissions for granting read-only access

IAM permissions for Infrastructure application manager persona

The following policy grants necessary permissions required for the Infrastructure application manager persona.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "InfrastructureApplicationManager",
      "Effect": "Allow",
      "Action": [
        "resiliencehub:AddDraftAppVersionResourceMappings",
        "resiliencehub:CreateAppVersionAppComponent",
        "resiliencehub:CreateAppVersionResource",
        "resiliencehub:CreateRecommendationTemplate",
        "resiliencehub:DeleteAppAssessment",
        "resiliencehub:DeleteAppInputSource",
        "resiliencehub:DeleteAppVersionAppComponent",
        "resiliencehub:DeleteAppVersionResource",
        "resiliencehub:DeleteRecommendationTemplate",
        "resiliencehub:Describe*",
        "resiliencehub:List*",
        "resiliencehub:PublishAppVersion",
        "resiliencehub:PutDraftAppVersionTemplate",
        "resiliencehub:RemoveDraftAppVersionResourceMappings",
        "resiliencehub:ResolveAppVersionResources",
        "resiliencehub:StartAppAssessment",
        "resiliencehub:TagResource",
        "resiliencehub:UntagResource",
        "resiliencehub:UpdateAppVersion",
        "resiliencehub:UpdateAppVersionAppComponent",
        "resiliencehub:UpdateAppVersionResource"
      ],
```

```
      "Resource": "*"
    }
  ]
}
```

IAM permissions for Business continuity manager persona

The following policy grants necessary permissions required for the Business continuity manager persona.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "BusinessContinuityManager",
      "Effect": "Allow",
      "Action": [
        "resiliencehub:CreateResiliencyPolicy",
        "resiliencehub:DeleteResiliencyPolicy",
        "resiliencehub:Describe*",
        "resiliencehub:List*",
        "resiliencehub:ResolveAppVersionResources",
        "resiliencehub:TagResource",
        "resiliencehub:UntagResource",
        "resiliencehub:UpdateAppVersion",
        "resiliencehub:UpdateAppVersionAppComponent",
        "resiliencehub:UpdateAppVersionResource",
        "resiliencehub:UpdateResiliencyPolicy"
      ],
      "Resource": "*"
    }
  ]
}
```

IAM permissions for Application owner persona

The following policy grants necessary permissions required for the Application owner persona.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "ApplicationOwner",
      "Effect": "Allow",
      "Action": [
        "resiliencehub:AddDraftAppVersionResourceMappings",
        "resiliencehub:BatchUpdateRecommendationStatus",
        "resiliencehub:CreateApp",
        "resiliencehub:CreateAppVersionAppComponent",
        "resiliencehub:CreateAppVersionResource",
        "resiliencehub:CreateRecommendationTemplate",
        "resiliencehub:CreateResiliencyPolicy",
        "resiliencehub:DeleteApp",
        "resiliencehub:DeleteAppAssessment",
        "resiliencehub:DeleteAppInputSource",
        "resiliencehub:DeleteAppVersionAppComponent",
        "resiliencehub:DeleteAppVersionResource",
        "resiliencehub:DeleteRecommendationTemplate",
        "resiliencehub:DeleteResiliencyPolicy",
        "resiliencehub:Describe*",
        "resiliencehub:ImportResourcesToDraftAppVersion",
        "resiliencehub:List*",
        "resiliencehub:PublishAppVersion",
        "resiliencehub:PutDraftAppVersionTemplate",
        "resiliencehub:RemoveDraftAppVersionResourceMappings",
        "resiliencehub:ResolveAppVersionResources",
        "resiliencehub:StartAppAssessment",
        "resiliencehub:TagResource",
        "resiliencehub:UntagResource",
        "resiliencehub:UpdateApp",
        "resiliencehub:UpdateAppVersion",
        "resiliencehub:UpdateAppVersionAppComponent",
        "resiliencehub:UpdateAppVersionResource",
        "resiliencehub:UpdateResiliencyPolicy"
      ],
      "Resource": "*"
    }
  ]
```

```
}
```

IAM permissions for granting read-only access

The following policy grants necessary permissions required for read-only access.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "ReadOnly",
      "Effect": "Allow",
      "Action": [
        "resiliencehub:Describe*",
        "resiliencehub:List*",
        "resiliencehub:ResolveAppVersionResources"
      ],
      "Resource": "*"
    }
  ]
}
```

Importing Terraform state file into AWS Resilience Hub

AWS Resilience Hub supports importing Terraform state files that are encrypted using server-side encryption (SSE) with Amazon Simple Storage Service managed keys (SSE-S3) or with AWS Key Management Service managed keys (SSE-KMS). If your Terraform state files are encrypted using customer-provided encryption keys (SSE-C), you will not be able to import them using AWS Resilience Hub.

Importing Terraform state files into AWS Resilience Hub requires the following IAM policies depending on where your state file is located.

Importing Terraform state files from an Amazon S3 bucket located in the primary account

The following Amazon S3 bucket policy and IAM policy are required to allow AWS Resilience Hub read access to your Terraform state files located in an Amazon S3 bucket on the primary account.

- Bucket policy – A bucket policy on the target Amazon S3 bucket, which is located in the primary account. For more information, see the following example.
- Identity policy – The associated identity policy for the Invoker role defined for this application, or the AWS current IAM role AWS Resilience Hub on the primary AWS account. For more information, see the following example.

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::<s3-bucket-name>/<path-to-state-file>"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::<s3-bucket-name>"
    }
  ]
}
```

Note

If you are using the `AWSResilienceHubAsssessmentExecutionPolicy` managed policy, `ListBucket` permission is not required.

Note

If your Terraform state files are encrypted using KMS, you must add the following `kms:Decrypt` permission.

```
{
      "Effect": "Allow",
      "Action": [
              "kms:Decrypt",
      ],
```

```
        "Resource": "<arn_of_kms_key>"
}
```

Importing Terraform state files from an Amazon S3 bucket located in a secondary account

- Bucket policy – A bucket policy on the target Amazon S3 bucket, which is located in one of the secondary accounts. For more information, see the following example.
- Identity policy – The associated identity policy for the AWS account role, which is running AWS Resilience Hub on the primary AWS account. For more information, see the following example.

> **Note**
>
> If you are using the `AWSResilienceHubAsssessmentExecutionPolicy` managed policy, `ListBucket` permission is not required.

> **Note**
>
> If your Terraform state files are encrypted using KMS, you must add the following `kms:Decrypt` permission.

```
{
        "Effect": "Allow",
        "Action": [
                "kms:Decrypt",
        ],
        "Resource": "<arn_of_kms_key>"
}
```

Enabling AWS Resilience Hub access to your Amazon Elastic Kubernetes Service cluster

AWS Resilience Hub assesses the resiliency of an Amazon Elastic Kubernetes Service (Amazon EKS) cluster by analyzing the infrastructure of your Amazon EKS cluster. AWS Resilience Hub

uses Kubernetes role-based access control (RBAC) configuration to assess other Kubernetes (K8s) workload, which are deployed as a part of the Amazon EKS cluster. For AWS Resilience Hub to query your Amazon EKS cluster for analyzing and assessing the workload, you must complete the following:

- Create or use an existing AWS Identity and Access Management (IAM) role in the same account as the Amazon EKS cluster.
- Enable IAM user and role access to your Amazon EKS cluster and grant additional read-only permissions to K8s resources inside the Amazon EKS cluster. For more information about enabling IAM user and role access to your Amazon EKS cluster, see Enabling IAM user and role access to your cluster - Amazon EKS.

Access to your Amazon EKS cluster using IAM entities are enabled by the AWS IAM Authenticator for Kubernetes, which runs on the Amazon EKS control plane. The Authenticator obtains the configuration information from `aws-auth ConfigMap`.

Note

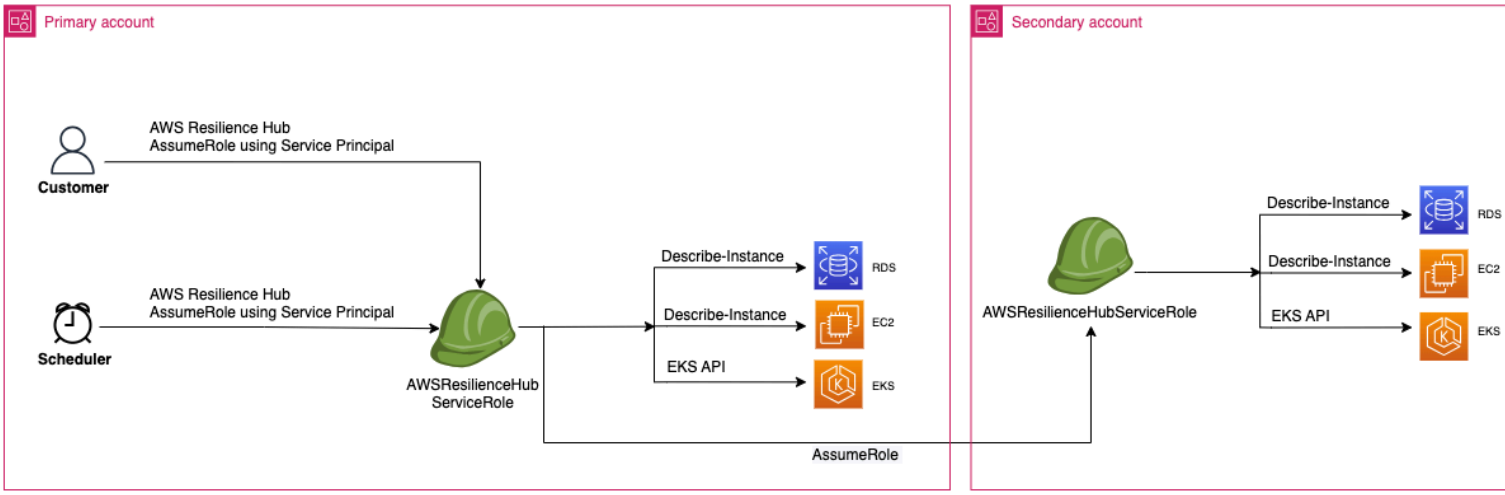
- For more information about all the `aws-auth ConfigMap` settings, see Full Configuration Format on GitHub.
- For more information about different IAM identities, see Identities (Users, Groups, and Roles) in the IAM User Guide.
- For more information about Kubernetes role-based access control (RBAC) configuration, see Using RBAC Authorization.

AWS Resilience Hub queries resources inside your Amazon EKS cluster using an IAM role in your account. For AWS Resilience Hub to access resources within your Amazon EKS cluster, the IAM role used by AWS Resilience Hub must be mapped to a Kubernetes group with sufficient read-only permissions to resources inside your Amazon EKS cluster.

AWS Resilience Hub allows to access your Amazon EKS cluster resources by using one of the following IAM role options:

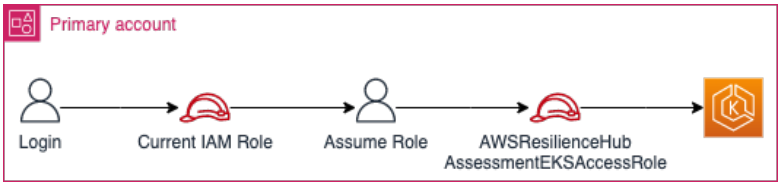
- If your application is configured to use role-based access for accessing resources, the invoker role or secondary account role passed to AWS Resilience Hub while creating an application will be used for accessing your Amazon EKS cluster during assessment.

The following conceptual diagram shows how AWS Resilience Hub accesses Amazon EKS clusters when the application is configured as a role-based application.

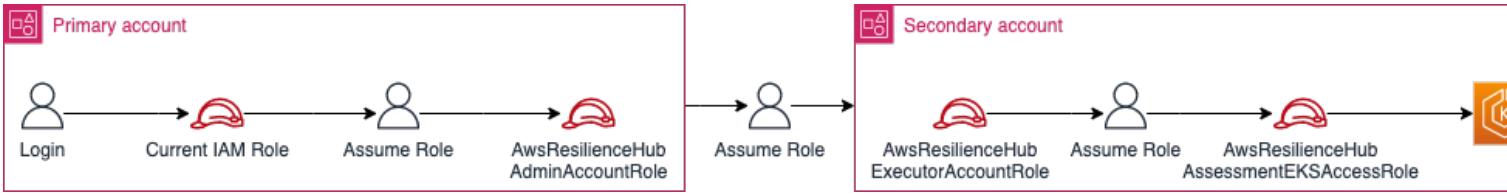


- If your application is configured to use the current IAM user for accessing resource, you must create a new IAM role with the name `AwsResilienceHubAssessmentEKSAccessRole` in the same account as that of the Amazon EKS cluster. This IAM role will then be used for accessing your Amazon EKS cluster.

  The following conceptual diagram shows how AWS Resilience Hub accesses Amazon EKS clusters deployed in your primary account when the application is configured to use the current IAM user permissions.



  The following conceptual diagram shows how AWS Resilience Hub accesses Amazon EKS clusters deployed on a secondary account when the application is configured to use the current IAM user permissions.



Granting AWS Resilience Hub access to resources in your Amazon EKS cluster

AWS Resilience Hub allows you to access resources located on Amazon EKS clusters provided you have configured the required permissions.

To grant required permissions to AWS Resilience Hub for discovering and assessing resources within Amazon EKS cluster

1. Configure an IAM role to access Amazon EKS cluster.

   If you have configured your application using role-based access, you can skip this step and proceed to step 2 and use the role that you had used for creating the application. For more information about how AWS Resilience Hub uses IAM roles, see the section called “How AWS Resilience Hub works with IAM”.

   If you have configured your application using current IAM user permissions, you must create `AwsResilienceHubAssessmentEKSAccessRole` IAM role in the same account as that of the Amazon EKS cluster. This IAM role will then be used while accessing your Amazon EKS cluster.

   While importing and assessing your application, AWS Resilience Hub uses an IAM role to access the resources in your Amazon EKS cluster. This role should be created in the same account as your Amazon EKS cluster and it will be mapped with a Kubernetes group that includes the permissions required by AWS Resilience Hub to assess your Amazon EKS cluster.

   If your Amazon EKS cluster is in the same account as the AWS Resilience Hub calling account, the role should be created using the following IAM trust policy. In this IAM trust policy, `caller_IAM_role` is used in the current account to call the APIs for AWS Resilience Hub.

   > **Note**
   >
   > The `caller_IAM_role` is the role that is associated with your AWS user account.

   If your Amazon EKS cluster is in a cross account (a different account than the AWS Resilience Hub calling account), you must create the `AwsResilienceHubAssessmentEKSAccessRole` IAM role using the following IAM trust policy:

> **Note**
>
> As a prerequisite, to access Amazon EKS cluster that is deployed in a different account than the AWS Resilience Hub user's account, you must configure multi-account access. For more information, see

2. Create `ClusterRole` and `ClusterRoleBinding` (or `RoleBinding`) roles for AWS Resilience Hub application.

   Creating `ClusterRole` and `ClusterRoleBinding` will grant the required read-only permissions for AWS Resilience Hub to analyze and assess resources that are a part of the certain namespaces in your Amazon EKS cluster.

   AWS Resilience Hub enables you to limit its access to your namespaces for generating resiliency assessments by completing one of the following:

   a. Grant read access across all namespaces to AWS Resilience Hub application.

      For AWS Resilience Hub to assess the resiliency of resources across all the namespaces within an Amazon EKS cluster, you must create the following `ClusterRole` and `ClusterRoleBinding`.

      - `resilience-hub-eks-access-cluster-role` (`ClusterRole`) – Defines the permissions required by AWS Resilience Hub to assess your Amazon EKS cluster.
      - `resilience-hub-eks-access-cluster-role-binding` (`ClusterRoleBinding`) – Defines a group named `resilience-hub-eks-access-group` in your Amazon EKS cluster granting its users, the required permissions to run resiliency assessments in AWS Resilience Hub.

      The template to grant read access across all namespaces to AWS Resilience Hub application is as follows:

```
cat << EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: resilience-hub-eks-access-cluster-role
```

```
rules:
- apiGroups:
    - ""
  resources:
    - pods
    - replicationcontrollers
    - nodes
  verbs:
    - get
    - list
- apiGroups:
    - apps
  resources:
    - deployments
    - replicasets
  verbs:
    - get
    - list
- apiGroups:
    - policy
  resources:
    - poddisruptionbudgets
  verbs:
    - get
    - list
- apiGroups:
    - autoscaling.k8s.io
  resources:
    - verticalpodautoscalers
  verbs:
    - get
    - list
- apiGroups:
    - autoscaling
  resources:
    - horizontalpodautoscalers
  verbs:
    - get
    - list
- apiGroups:
    - karpenter.sh
  resources:
    - provisioners
    - nodepools
```

```
  verbs:
    - get
    - list
- apiGroups:
    - karpenter.k8s.aws
  resources:
    - awsnodetemplates
    - ec2nodeclasses
  verbs:
    - get
    - list
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: resilience-hub-eks-access-cluster-role-binding
subjects:
  - kind: Group
    name: resilience-hub-eks-access-group
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: resilience-hub-eks-access-cluster-role
  apiGroup: rbac.authorization.k8s.io
---
EOF
```

b. Granting AWS Resilience Hub the access to read specific namespaces.

You can limit AWS Resilience Hub to access resources within a specific set of namespaces using `RoleBinding`. To achieve this, you must create the following roles:

- `ClusterRole` – For AWS Resilience Hub to access the resources in specific namespaces within an Amazon EKS cluster and assess its resiliency, you must create the following `ClusterRole` roles.
  - `resilience-hub-eks-access-cluster-role` – Specifies the necessary permissions to assess the resources within specific namespaces.
  - `resilience-hub-eks-access-global-cluster-role` – Specifies the necessary permissions to assess cluster-scoped resources, which are not associated to a specific namespace, within your Amazon EKS clusters. AWS Resilience Hub requires

permissions to access cluster-scoped resources (such as nodes) on your Amazon EKS cluster to assess the resiliency of your application.

The template to create `ClusterRole` role is as follows:

```
cat << EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: resilience-hub-eks-access-cluster-role
rules:
  - apiGroups:
      - ""
    resources:
      - pods
      - replicationcontrollers
    verbs:
      - get
      - list
  - apiGroups:
      - apps
    resources:
      - deployments
      - replicasets
    verbs:
      - get
      - list
  - apiGroups:
      - policy
    resources:
      - poddisruptionbudgets
    verbs:
      - get
      - list
  - apiGroups:
      - autoscaling.k8s.io
    resources:
      - verticalpodautoscalers
    verbs:
      - get
      - list
  - apiGroups:
```

```
      - autoscaling
    resources:
      - horizontalpodautoscalers
    verbs:
      - get
      - list

---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: resilience-hub-eks-access-global-cluster-role
rules:
  - apiGroups:
      - ""
    resources:
      - nodes
    verbs:
      - get
      - list
  - apiGroups:
      - karpenter.sh
    resources:
      - provisioners
      - nodepools
    verbs:
      - get
      - list
  - apiGroups:
      - karpenter.k8s.aws
    resources:
      - awsnodetemplates
      - ec2nodeclasses
    verbs:
      - get
      - list

---
EOF
```

- `RoleBinding` role – This role grants the required permissions for AWS Resilience Hub to access resources within specific namespaces. That is, you must create `RoleBinding`

role in each namespace to enable AWS Resilience Hub to access resources within the given namespace.

> **Note**
>
> If you are using `ClusterAutoscaler` for autoscaling, you must additionally create `RoleBinding` in the `kube-system`. This is necessary to assess your `ClusterAutoscaler`, which is a part of the `kube-system` namespace.
> By doing this, you will grant AWS Resilience Hub the required permissions to assess resources inside the `kube-system` namespace while assessing your Amazon EKS cluster.

The template to create `RoleBinding` role is as follows:

```
cat << EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: resilience-hub-eks-access-cluster-role-binding
  namespace: <namespace>
subjects:
  - kind: Group
    name: resilience-hub-eks-access-group
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: resilience-hub-eks-access-cluster-role
  apiGroup: rbac.authorization.k8s.io

---
EOF
```

- `ClusterRoleBinding` role – This role grants the required permissions for AWS Resilience Hub to access cluster-scoped resources.

  The template to create `ClusterRoleBinding` role is as follows:

```
cat << EOF | kubectl apply -f -
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: resilience-hub-eks-access-global-cluster-role-binding
subjects:
  - kind: Group
    name: resilience-hub-eks-access-group
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: resilience-hub-eks-access-global-cluster-role
  apiGroup: rbac.authorization.k8s.io

---
EOF
```

3. Update the `aws-auth ConfigMap` to map the `resilience-hub-eks-access-group` with the IAM role that is used for accessing Amazon EKS cluster.

   This step creates a mapping between the IAM role used in step 1 with the Kubernetes group created in step 2. This mapping grants permissions to IAM roles for accessing resources inside the Amazon EKS cluster.

   Note

   - `ROLE-NAME` refers to the IAM role that is used for accessing Amazon EKS cluster.
     - If your application is configured to use role-based access, the role should be either the invoker role or secondary account role that is passed to AWS Resilience Hub while creating the application.
     - If your application is configured to use the current IAM user for accessing resources, it must be the `AwsResilienceHubAssessmentEKSAccessRole`.
   - `ACCOUNT-ID` should be the AWS account ID of the Amazon EKS cluster.

   You can create the `aws-auth ConfigMap` using one of the following ways:

- Using `eksctl`

  Use the following command to update the `aws-auth` `ConfigMap`:

  ```
  eksctl create iamidentitymapping \
   --cluster <cluster-name> \
   --region=<region-code> \
   --arn arn:aws:iam::<ACCOUNT-ID>:role/<ROLE-NAME>\
   --group resilience-hub-eks-access-group \
   --username AwsResilienceHubAssessmentEKSAccessRole
  ```

- You can manually edit `aws-auth` `ConfigMap` by adding the IAM role details to `mapRoles` section of the `ConfigMap` under data. Use the following command to edit the `aws-auth` `ConfigMap`.

  ```
  kubectl edit -n kube-system configmap/aws-auth
  ```

  `mapRoles` section consists of the following parameters:

  - `rolearn` – The Amazon Resource Name (ARN) of the IAM role to be added.
    - ARN Syntax – `arn:aws:iam::<ACCOUNT-ID>:role/<ROLE-NAME>`.
  - `username` – The username within Kubernetes to be mapped to the IAM role (`AwsResilienceHubAssessmentEKSAccessRole`).
  - `groups` – The group names should match the group names created in **Step 2** (`resilience-hub-eks-access-group`).

  > **Note**
  >
  > If `mapRoles` section does not exist, you must manually add this section.

  Use the following template to add the IAM role details to `mapRoles` section of the `ConfigMap` under data.

  ```
    - groups:
      - resilience-hub-eks-access-group
      rolearn: arn:aws:iam::<ACCOUNT-ID>:role/<ROLE-NAME>
      username: AwsResilienceHubAssessmentEKSAccessRole
  ```

Enabling AWS Resilience Hub to publish to your Amazon Simple Notification Service topics

This section explains about how to enable AWS Resilience Hub to publish notifications about the application to your Amazon Simple Notification Service (Amazon SNS) topics. To push notifications to an Amazon SNS topic, ensure that you have the following:

- An active AWS Resilience Hub application.
- An existing Amazon SNS topic to which AWS Resilience Hub must send notifications. For more information about creating an Amazon SNS topic, see Creating an Amazon SNS topic.

To enable AWS Resilience Hub to publish notifications to your Amazon SNS topic, you must update the access policy of the Amazon SNS topic with the following:

> **Note**
>
> When you use AWS Resilience Hub to publish messages from opt-in Regions to topics located in Regions that are enabled by default, you must modify the resource policy created for the Amazon SNS topic. Change the value of principal from `resiliencehub.amazonaws.com` to `resiliencehub.<opt-in-region>.amazonaws.com`.

If you are using a Server Side Encrypted (SSE) Amazon SNS topic, you must ensure that AWS Resilience Hub has the `Decrypt` and `GenerateDataKey*` access to the Amazon SNS encryption key.

To provide `Decrypt` and `GenerateDataKey*` access to AWS Resilience Hub, you must include the following permissions to AWS Key Management Service access policy.

Limiting permissions to include or exclude AWS Resilience Hub recommendations

AWS Resilience Hub enables you to restrict permissions to include or exclude recommendations per application. You can restrict permissions to include or exclude recommendations per application using the following IAM trust policy. In this IAM trust policy, `caller_IAM_role` (associated with your AWS user account) is used in the current account to call the APIs for AWS Resilience Hub.

Infrastructure security in AWS Resilience Hub

As a managed service, AWS Resilience Hub is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes white paper.

You use AWS published API calls to access AWS Resilience Hub through the network. Clients must support Transport Layer Security (TLS) 1.2 or later. We recommend TLS 1.3 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Resilience Checks for AWS services

This chapter provides the details of various resilience checks performed by AWS Resilience Hub for supported AWS services to ensure that the resiliency postures of applications are not affected. These checks estimate the recovery time objective (RTO) and recovery point objective (RPO) against the values defined in the resilience policy for each Application Component (AppComponent). The assessments encompass different types of disruptions, that is, Application, Infrastructure failures, AZ outages, and Regional failures. However, to run these checks you must provide relevant IAM permissions to AWS Resilience Hub for allowing it to access your resources. To learn more about the required IAM permissions to allow AWS Resilience Hub to access your resources and perform the resilience checks in this chapter, see AWS managed policies for AWS Resilience Hub.

AWS services

- Amazon Elastic File System
- Amazon Relational Database Service and Amazon Aurora
- Amazon Simple Storage Service
- Amazon DynamoDB
- Amazon Elastic Compute Cloud
- Amazon EBS
- AWS Lambda

- Amazon Elastic Kubernetes Service
- Amazon Simple Notification Service
- Amazon Simple Queue Service
- Amazon Elastic Container Service
- Elastic Load Balancing
- Amazon API Gateway
- Amazon DocumentDB
- NAT Gateway
- Amazon Route 53
- Amazon Application Recovery Controller (ARC)
- Amazon FSx for Windows File Server
- AWS Step Functions
- Amazon ElastiCache (Redis OSS)

Amazon Elastic File System

This section lists all the resilience checks and recommendations that are specific to Amazon Elastic File System. For more information about Amazon Elastic File System, see the Amazon Elastic File System documentation.

Filesystem type

AWS Resilience Hub checks filesystem type: Regional or One Zone. The filesystem type affects its resiliency in the event of Infrastructure or AZ disruptions. For more information about filesystem types, see Availability and durability of Amazon EFS file systems.

Filesystem Backup

AWS Resilience Hub checks if an AWS Backup plan is defined for the deployed filesystem. Additionally, it verifies if the `Cross-Region` backup option is enabled, ensuring coverage for Region-level disruptions if required by your policy.

Data Replication

AWS Resilience Hub checks if an in-Region or cross-Region Amazon EFS data replication is defined for the deployed filesystem. Amazon EFS data replication helps to improve estimated RTO and

estimated RPO at Application, Infrastructure, AZ, and Region levels. Additionally, AWS Resilience Hub checks if it is combined with an in-Region AWS Backup to enable filesystem resiliency in the event of application disruption.

Amazon Relational Database Service and Amazon Aurora

This section lists all the resilience checks and recommendations that are specific for Amazon Relational Database Service and Amazon Aurora. For more information about Amazon Relational Database Service and Amazon Aurora, see Amazon Relational Database Service documentation.

Single-AZ deployment

AWS Resilience Hub checks if the database is deployed as a single instance and if determined, it indicates that it does not support secondary instance and read replica.

Multi-AZ deployment

AWS Resilience Hub checks if the database is deployed either with secondary instance or read replicas. If the database is deployed with read replica, AWS Resilience Hub validates if it is deployed in a different AZ to allow failover in the event of an AZ disruption.

Backup

AWS Resilience Hub checks if the following backup capabilities are applied on a deployed database instance.

- AWS Backup plan with automatic backup option
- AWS Backup plan with cross-Region backup copy if it is required by your policy
- Manual snapshots for 3rd party backup systems

Cross-Region failover

AWS Resilience Hub checks RTO and RPO targets that are defined in the resiliency policy to recover from Regional disruption. Additionally, AWS Resilience Hub can identify following cross-Region architectures to cover for Regional disruption:

- An in-Region backup with a copy of a cross-Region snapshot
- A read replica in another Region

- An Amazon Aurora global database with a secondary cluster in another Region
- An Amazon Aurora global database with a headless secondary cluster in another Region

Faster in-Region failover

AWS Resilience Hub checks RTO and RPO targets defined in the resiliency policy during infrastructure or AZ disruptions. Additionally, AWS Resilience Hub can identify the following in-Region architectures to cover for Application, Infrastructure and AZ disruptions:

- An In-Region backup
- A read replica in a different AZ
- An Aurora cluster with a read replica in another AZ
- A Multi-AZ instance of Amazon Relational Database Service (Amazon RDS)
- An Amazon RDS Multi-AZ cluster
- A single instance of Amazon RDS with a read replica in another AZ

Amazon Simple Storage Service

This section lists all the resilience checks and recommendations that are specific for Amazon Simple Storage Service (Amazon S3). For more information about Amazon S3, see Amazon S3 documentation.

Versioning

AWS Resilience Hub verifies if an Amazon S3 bucket is configured with versioning enabled.

Scheduled backup

AWS Resilience Hub checks if an AWS Backup plan is defined for the deployed Amazon Simple Storage Service (Amazon S3) bucket. Additionally, it also checks if cross-Region backup option is enabled if your policy requires coverage for Region-level disruptions.

Point-in-time recovery

AWS Resilience Hub checks if point-in-time recovery (PITR) is required by your resiliency policy's RPO target. However, cross-Region backup is not supported for PITR. Hence, you use an existing scheduled AWS Backup plan with cross-Region backup option enabled, or create a new one.

Data replication

AWS Resilience Hub checks if a Same Region Replication (SRR) and Cross Region Replication (CRR) is defined for the deployed Amazon S3 bucket. Amazon S3 data replication improves estimated workload RTO and estimated workload RPO at Application, Infrastructure, AZ, and Region level. Additionally, it also protects from physical deletion of object because deletion of an object version is not replicated to the target Amazon S3 bucket. Additionally, based on the RTO targets defined in your resiliency policy, AWS Resilience Hub checks if Amazon S3 Replication Time Control (S3 RTC) should be enabled or not. This billable feature replicates 99.99 percent of source bucket objects within 15 minutes.

- AWS Backup plan with automatic backup option
- AWS Backup plan with cross-Region backup copy if it is required by your policy
- Manual snapshots for 3rd party backup systems

Amazon DynamoDB

This section lists all the resilience checks and recommendations that are specific for Amazon DynamoDB. For more information about Amazon DynamoDB, see Amazon DynamoDB documentation.

Scheduled backup

AWS Resilience Hub checks if a backup is already defined for the deployed table. Additionally, it also checks if cross-Region backup should be configured for your policy if it requires coverage for Region-level disruptions.

Point-in-time recovery

AWS Resilience Hub checks if point-in-time recovery (PITR) is required according to your resiliency policy's RPO target. However, cross-Region backup is not supported for PITR. Hence, you use an existing scheduled AWS Backup plan with cross-Region backup option enabled, or create a new one.

Global table

AWS Resilience Hub checks if the deployed Amazon DynamoDB table is defined as a Global Table with one or more replicas in other Regions. Setting up Global Table improves estimated workload RTO and estimated workload RPO at Region level, and also provides a capability to work in active-

active or active-passive multi-Region modes. AWS Backup or Amazon DynamoDB PITR can be used in one of the Regions to handle application disruptions.

Amazon Elastic Compute Cloud

This section lists all the resilience checks and recommendations that are specific for Amazon Elastic Compute Cloud. For more information about Amazon Elastic Compute Cloud, see Amazon Elastic Compute Cloud documentation.

Stateful instance

AWS Resilience Hub identifies an Amazon EC2 instance as a stateful instance if one of the following criteria is met:

- If `DeleteOnTermination` attribute is set to false for at least one Amazon Elastic Block Store (Amazon EBS) volume that is attached to this instance.
- If Amazon Data Lifecycle Manager or an AWS Backup plan is attached to the Amazon EC2 instance or at least one Amazon EBS volume.
- If AWS Elastic Disaster Recovery is used to replicate your Amazon EC2 instance storage volumes.

> **Note**
>
> If an Amazon EC2 instance doesn't meet the any of the above criteria, AWS Resilience Hub treats it as a stateless Amazon EC2 instance.

Auto Scaling groups

AWS Resilience Hub checks for a group of stateless Amazon EC2 instances. If discovered, it is recommended to orchestrate the same using Auto Scaling groups (ASG) with Multi-AZ configuration. If an existing ASG is identified, ARH will verify if it is configured across multiple Availability Zones. If ASG is also defined using spot Amazon EC2 instances only, it is recommended to augment its capacity with on-demand Amazon EC2 instances to improve the resiliency when spot Amazon EC2 instances are unavailable.

Amazon EC2 Fleet

AWS Resilience Hub identifies Amazon EC2 Fleet and verifies if it is defined as Multi-AZ deployment and also if it uses spot Amazon EC2 instances only. Defining an Amazon EC2 Fleet as Multi-AZ

deployment will improve its resiliency in the event of an AZ disruption. Augmenting an Amazon EC2 Fleet with on-demand instances will improve its resiliency when spot instances are unavailable.

Amazon EBS

This section lists all the resilience checks and recommendations that are specific to Amazon EBS. For more information about Amazon EBS, see Amazon EBS documentation.

Scheduled backup

AWS Resilience Hub checks if either or both the following are defined for your Amazon EBS volumes.

- A backup rule for specific Amazon EBS volume attached to your Amazon EC2 instance.
- A backup rule to create Amazon EBS-backed AMI to your Amazon EC2 instance.
- Manual snapshots for 3rd party backup systems.

Additionally, if your policy requires coverage for Region-level disruptions, AWS Resilience Hub checks if your backup rule has cross-Region backup option enabled.

Data backup and replication

AWS Resilience Hub identifies an Amazon EBS volume is considered a stateful volume if one of the following criteria is met:

- If `DeleteOnTermination` attribute is set to false for this Amazon EBS volume.
- If Amazon Data Lifecycle Manager or an AWS Backup plan is associated with either this Amazon EBS volume or the Amazon EC2 instance it is attached to.
- If AWS Elastic Disaster Recovery is used to replicate your Amazon EC2 instance storage volumes.

AWS Lambda

This section lists all the resilience checks and recommendations that are specific to AWS Lambda. For more information about AWS Lambda, see AWS Lambda documentation.

Customer Amazon VPC Access

AWS Resilience Hub identifies an AWS Lambda function connected to the VPC. Connecting AWS Lambda to subnets in different AZs of your Amazon VPC allows function resiliency in case of an AZ disruption.

Dead-letter queue

AWS Resilience Hub checks if an AWS Lambda function has a dead-letter queue (DLQ) attached to it for storing failed requests. Attaching a DLQ to AWS Lambda function allows to prevent the data loss of requests and retry to process the failed requests at a later stage.

Amazon Elastic Kubernetes Service

This section lists all the resilience checks and recommendations that are specific to Amazon Elastic Kubernetes Service (Amazon EKS). For more information about Amazon EKS, see Amazon EKS documentation.

Multi-AZ deployment

AWS Resilience Hub identifies if pod deployment is running on multiple worker nodes in multiple AZs. An additional Amazon EKS cluster in another Region is required if your resiliency policy requires coverage in the event of Regional disruption. This additional Amazon EKS cluster is also verified for pod deployments that are distributed between multiple worker nodes in multiple AZs.

Deployment vs. ReplicaSet

AWS Resilience Hub checks if you are using ReplicaSets or pod objects instead of deployment. Replacing ReplicaSets or pod objects with deployment simplifies the pod updates to a new version of the software and includes other useful features.

Deployment maintenance

AWS Resilience Hub checks if the following best practices are used for deployment:

- Using Pod Disruption Budget (PDB) – Using PDB makes it possible to improve the availability by setting a limit on the number of pods in the workload that can be disrupted at any given time.
- Replacing self-managed node groups with Amazon EKS managed node groups – This replacement simplifies worker node image updates during maintenance.

- Supporting dynamic CPU and memory requests per deployment – These requests help Kubernetes to select a node that fits the needs of a pod.
- Configuring liveness and readiness probes for all the containers – Configuring liveness probes help to improve the resiliency by restarting the non-functional pods. Configuring readiness probes make it possible to improve the availability by diverting the traffic away from busy pods.
- Configuring Karpenter, Cluster Autoscaler, or AWS Fargate – These configurations allow Amazon EKS cluster's infrastructure to grow and meet the workload demands.
- Configuring Horizontal Pod Autoscaler – This configuration helps Amazon EKS cluster to automatically scale the workload to meet request processing demand.

Amazon Simple Notification Service

This section lists all the resilience checks and recommendations that are specific to Amazon Simple Notification Service (Amazon SNS). For more information about Amazon SNS, see Amazon SNS documentation.

Topic subscriptions

AWS Resilience Hub checks if Amazon SNS topic has at least 1 subscription attached to it for ensuring that incoming messages are not lost.

Amazon Simple Queue Service

This section lists all the resilience checks and recommendations that are specific to Amazon Simple Queue Service (Amazon SQS). For more information about Amazon SQS, see Amazon SQS documentation.

Dead-letter queue

AWS Resilience Hub checks if the Amazon SQS queue has a DLQ associated to it to handle messages that can't be delivered to subscribers successfully.

Amazon Elastic Container Service

This section lists all the resilience checks and recommendations that are specific to Amazon Elastic Container Service (Amazon ECS). For more information about Amazon ECS, see Amazon ECS documentation.

Multi-AZ deployment

AWS Resilience Hub checks if Amazon ECS tasks or services are running in multiple AZs based on either Amazon EC2 or AWS Fargate launch types. An additional Amazon ECS cluster in another Region is required if your policy needs coverage for Regional disruption. The additional cluster is also verified for execution of tasks or services in multiple AZs.

Elastic Load Balancing

This section lists all the resilience checks and recommendations that are specific to Elastic Load Balancing. For more information about Elastic Load Balancing, see Elastic Load Balancing documentation.

Multi-AZ deployment

AWS Resilience Hub checks if Elastic Load Balancing is running in multiple AZs.

An additional Elastic Load Balancing in a different Region is required if your policy needs coverage for Regional disruption. The additional Elastic Load Balancing, located in a different Region, is also verified for its deployment in multiple AZs.

Amazon API Gateway

This section lists all the resilience checks and recommendations that are specific to Amazon API Gateway. For more information about Amazon API Gateway, see Amazon API Gateway documentation.

Cross-Region deployment

If your policy needs to consider Regional disruption, AWS Resilience Hub will check if there is an additional deployment of Amazon API Gateway API resource in a different Region.

Private API Multi-AZ deployment

AWS Resilience Hub checks if your API is defined as private within Amazon API Gateway. Private APIs should receive traffic through Amazon VPC interface endpoint that is deployed to multiple AZs.

Amazon DocumentDB

This section lists all the checks and recommendations that are specific to Amazon DocumentDB. For more information about Amazon DocumentDB, see Amazon DocumentDB documentation.

Multi-AZ deployment

AWS Resilience Hub checks if Amazon DocumentDB cluster is deployed in multiple AZs. An additional secondary Amazon DocumentDB cluster is required in a different Region if your policy requires coverage for Regional disruption. The additional Amazon DocumentDB cluster, located in a different Region, is also verified for its execution in multiple AZs.

Elastic cluster and Multi-AZ deployment

AWS Resilience Hub checks if Amazon DocumentDB Elastic cluster shards are using read replicas that are deployed in different AZs.

Elastic cluster and Manual snapshots

AWS Resilience Hub checks if manual snapshots are regularly created for an Amazon DocumentDB Elastic cluster. Manual snapshots allow longer persistence and provides flexibility in setting the snapshot frequency to suit your business needs.

NAT Gateway

This section lists all the checks and recommendations that are specific to NAT Gateway. For more information about NAT Gateways, see NAT Gateways.

Multi-AZ deployment

AWS Resilience Hub checks if NAT Gateway is deployed in multiple AZs. An additional NAT Gateway deployment is required in a different Region if your policy requires coverage for Regional disruption. The additional NAT Gateway, located in a different Region, is also verified for its deployment in multiple AZs.

Amazon Route 53

This section lists all the checks and recommendations that are specific to Amazon Route 53. For more information about Amazon Route 53, see Amazon Route 53 documentation.

Multi-AZ deployment

AWS Resilience Hub checks if Amazon Route 53 hosted zone record is defined with multiple targets in the same Region and if these targets are deployed in multiple AZs. If your policy requires coverage for Regional disruption, AWS Resilience Hub checks if Amazon Route 53 hosted zone record is defined in multiple Regions with multiple targets per Region, and if these targets are deployed in multiple AZs.

Amazon Application Recovery Controller (ARC)

This section lists all the checks and recommendations that are specific to Amazon Application Recovery Controller (ARC) (ARC). For more information about ARC, see ARC documentation.

Multi-AZ deployment

AWS Resilience Hub checks if similar resources are deployed in multiple Regions and recommends as a best practice to define ARC readiness checks to increase their availability and readiness in the event of a Regional disruption. You will be notified that you will incur additional hourly charges.

Amazon FSx for Windows File Server

This section lists all the checks and recommendations that are specific to Amazon FSx for Windows File Server. For more information about Amazon FSx for Windows File Server, see Amazon FSx for Windows File Server documentation.

Filesystem type

AWS Resilience Hub checks the filesystem type: `Regional` or `One Zone`. Filesystem type affects its resiliency in the event of Infrastructure or AZ disruptions. For more information about filesystem types, see Amazon EFS.

Filesystem Backup

AWS Resilience Hub checks if an AWS Backup is defined for the deployed filesystem. Additionally, it also checks if `cross-Region backup` option is enabled if your policy requires coverage for Region-level disruptions.

Data Replication

AWS Resilience Hub checks if an in-Region or cross-Region scheduled AWS DataSync data replication task is defined for the deployed filesystem.

AWS DataSync scheduled data replication task can improve estimated workload RTO and estimated workload RPO at Infrastructure, AZ, and Region levels. Additionally, it could be combined with an in-Region AWS Backup to recover in the event of an application disruption.

AWS Step Functions

This section lists all the checks and recommendations that are specific to AWS Step Functions. For more information about AWS Step Functions, see AWS Step Functions documentation.

Versioning and alias

AWS Resilience Hub checks if AWS Step Functions workflow uses versioning and alias to improve the re-deployment time.

Cross-Region deployment

AWS Resilience Hub checks if AWS Step Functions workflow of the same workflow type is deployed in a different Region to recover in the event of a Regional disruption.

Amazon ElastiCache (Redis OSS)

This section lists all the checks and recommendations that are specific to Amazon ElastiCache (Redis OSS).

For more information about Amazon ElastiCache (Redis OSS), see Amazon ElastiCache documentation.

Single-AZ deployment

AWS Resilience Hub checks if Amazon ElastiCache (Redis OSS) cluster is deployed either as a single node or with all its nodes in a single Availability Zone.

Single-AZ deployment

AWS Resilience Hub validates if Amazon ElastiCache (Redis OSS) cluster is deployed as a replication group (for both Cluster Mode enabled and Cluster Mode Disabled clusters) across multiple Availability Zones to allow failover in the event of an Availability Zone disruption.

Cross-Region failover

AWS Resilience Hub checks RTO and RPO targets that are defined in the resiliency policy to recover from a Regional disruption. Additionally, AWS Resilience Hub can identify Amazon ElastiCache (Redis OSS) global datastore clusters deployed in multiple Regions.

Backup

AWS Resilience Hub checks if the following backup capabilities are applied on a deployed Amazon ElastiCache (Redis OSS) or self-designed cluster:

- Automatic backup
- Manual backup for 3rd party backup systems

AWS Resilience Hub will not recommend backup as a recovery method if you are not using backup. However, you can reset Cache layer in the event of data inconsistency and recreate the data from the primary storage.

Faster in-Region failover

AWS Resilience Hub checks RTO and RPO targets defined in the resiliency policy during infrastructure or AZ disruptions. Additionally, AWS Resilience Hub can identify the following in-Region architectures to recover from Infrastructure and AZ disruptions:

- Secondary standby node instance in a different Availability Zone for Cluster Mode Disabled type of Amazon ElastiCache (Redis OSS) cluster.
- Secondary standby node instance in a different Availability Zone per every shard for Cluster Mode Enabled type of Amazon ElastiCache (Redis OSS) cluster.

Working with other services

This section describes AWS services that interact with AWS Resilience Hub.

Topics

- AWS CloudFormation
- AWS CloudTrail
- AWS Systems Manager

- AWS Trusted Advisor

AWS CloudFormation

AWS Resilience Hub is integrated with AWS CloudFormation, a service that helps you to model and set up your AWS resources so that you can spend less time creating and managing your resources and infrastructure. You create a template that describes all the AWS resources that you want (such as AWS::ResilienceHub:::ResiliencyPolicy and AWS::ResilienceHub:::App), and CloudFormation provisions and configures those resources for you.

When you use CloudFormation, you can reuse your template to set up your AWS Resilience Hub resources consistently and repeatedly. Describe your resources one time, and then provision the same resources repeatedly in multiple AWS accounts and Regions.

AWS Resilience Hub and CloudFormation templates

To provision and configure resources for AWS Resilience Hub and related services, you must understand CloudFormation templates. Templates are formatted text files in JSON or YAML. These templates describe the resources that you want to provision in your CloudFormation stacks. If you're unfamiliar with JSON or YAML, you can use CloudFormation Designer to help you get started with CloudFormation templates. For more information, see What is CloudFormation Designer? in the *AWS CloudFormation User Guide*.

AWS Resilience Hub supports creating AWS::ResilienceHub:::ResiliencyPolicy and AWS::ResilienceHub:::App in CloudFormation. For more information, including examples of JSON and YAML templates for AWS::ResilienceHub:::ResiliencyPolicy and AWS::ResilienceHub:::App, see the AWS Resilience Hub resource type reference in the *AWS CloudFormation User Guide*.

You can use CloudFormation stacks to define AWS Resilience Hub applications. A stack lets you manage related resources as a single unit. A stack can contain all the resources that you need to run a web application, such as a web server or networking rules.

Learn more about CloudFormation

For more information about CloudFormation, see the following resources:

- AWS CloudFormation
- AWS CloudFormation User Guide

- CloudFormation API Reference
- AWS CloudFormation Command Line Interface User Guide

AWS CloudTrail

AWS Resilience Hub is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, a role, or an AWS service in AWS Resilience Hub. CloudTrail captures all API calls for AWS Resilience Hub as events. The calls that are captured include calls from the AWS Resilience Hub console and code calls to the AWS Resilience Hub API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Resilience Hub. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in Event history. Using the information collected by CloudTrail, you can determine the request that was made to AWS Resilience Hub, the IP address from which the request was made, who made the request, when it was made, and additional details.

For more information about CloudTrail, see the AWS CloudTrail User Guide.

AWS Systems Manager

AWS Resilience Hub works with Systems Manager to automate the steps of your SOPs by providing a number of SSM documents you can use as the basis for those SOPs.

AWS Resilience Hub provides you CloudFormation templates that contains the IAM roles required to run different Systems Manager documents, one role per document with permissions required for the specific document. After creating a stack with the CloudFormation template, it will setup the IAM roles and save metadata in Systems Manager parameter for the Systems Manager automation document to run for different recovery procedures.

For more information on using SOPs, see Managing standard operating procedures.

AWS Trusted Advisor

AWS Trusted Advisor is a centralized home of AWS best practice recommendations that helps you to identify, prioritize, and optimize your deployment on AWS. AWS Trusted Advisor inspects your AWS environment, and then makes recommendations through checks when opportunities exist to save money, improve system availability and performance, or help close security gaps. These checks are divided into multiple categories based on their purpose. For more information about different categories of checks in AWS Trusted Advisor, see the AWS Support User Guide.

AWS Trusted Advisor provides multiple high-level resiliency recommendations through resiliency checks for each application in AWS Resilience Hub under **Fault tolerance** category. **Fault tolerance** category lists all the checks that tests your applications to determine their resilience and reliability. These checks alert you when there are AppComponent failures and policy breaches that can cause resiliency risks and affect the application availability for business continuity. It also provides resiliency recommendations that will improve the chances to reduce these risks under **Recommended Action** section, which needs to be addressed in AWS Resilience Hub. For more insights about the recommendations for each application in the AWS Trusted Advisor, we recommend you to view the detailed recommendations provided in the AWS Resilience Hub.

AWS Trusted Advisor provides the following checks for each application in AWS Resilience Hub:

- **AWS Resilience Hub application resilience scores** – Checks the resiliency score of your applications from their latest assessment in AWS Resilience Hub and alerts you if their resiliency scores are below a specific value.

  Alert criteria

  - **Green** – Indicates that your application has a resiliency score of 70 and above.
  - **Yellow** – Indicates that your application has a resiliency score between 40 and 69.
  - **Red** – Indicates that your application has a resiliency score less than 40.

  Recommended action

  To improve the resiliency posture and obtain the best possible resiliency score for your application, run an assessment with the most recently updated version of your application resources and if applicable, implement the suggested operational recommendations. For more information about running, reviewing, and implementing assessments, reviewing and including/excluding operational recommendations, and implementing the same, see the following topics:

  - the section called “Running resiliency assessments in AWS Resilience Hub”
  - the section called “Reviewing assessments reports”
  - the section called “Reviewing resiliency recommendations”
  - the section called “Including or excluding operational recommendations”
- **AWS Resilience Hub application policy breached** – Checks if the AWS Resilience Hub applications meet the RTO and RPO targets you have set for an application and alerts you if the application do not meet the RTO and RPO targets.

  Alert criteria

- **Green** – Indicates that the application has a policy and the estimated workload RTO and estimated workload RPO meet the RTO and RPO targets.
- **Yellow** – Indicates that the application has a policy and has not been assessed.
- **Red** – Indicates that the application has a policy and the estimated workload RTO and estimated workload RPO does not meet the RTO and RPO targets.

Recommended action

To ensure that the estimated workload RTO and estimated workload RPO of your application still meet the defined RTO and RPO targets, run assessments regularly with the most recently updated version of your application resources. In addition, if you want to ensure that the resiliency policy of your application is not breached, we recommend you to review the assessment report and implement the suggested resiliency recommendations. For more information about enabling AWS Resilience Hub to run assessments on a daily basis on your behalf, running assessments, reviewing resiliency recommendations and implementing the same, see the following topics:

- the section called “Editing application resources” (To enable AWS Resilience Hub to run assessments on a daily basis on your behalf, complete the steps in **To edit drift notification settings of your application** procedure to select **Automatically assess daily** check box.)
- the section called “Running resiliency assessments in AWS Resilience Hub”
- the section called “Reviewing assessments reports”
- the section called “Reviewing resiliency recommendations”
- the section called “Including or excluding operational recommendations”

- **AWS Resilience Hub application assessment age** – Checks the last time since you had run an assessment for each of your applications in AWS Resilience Hub. It alerts you if you haven’t run an assessment for the specified number of days.

Alert criteria

- **Green** – Indicates that you have run an assessment for your application in the last 30 days.
- **Yellow** – Indicates that you have not run an assessment for your application in the last 30 days.

Recommended action

Run assessments regularly to manage and improve the resilience posture of your applications on AWS. If you want AWS Resilience Hub to assess your application on a daily basis on your behalf,

you can enable the same by the selecting the **Automatically assess this application daily** check box in AWS Resilience Hub drift notification. To select **Automatically assess this application daily** check box, complete the **To edit drift notification of your application** procedure in ???.

> **Note**
>
> This check determines the assessment age of only those applications that have been assessed at-least once in AWS Resilience Hub.

- **AWS Resilience Hub application component check** – Checks if an Application Component (AppComponent) in your application is unrecoverable. That is, if this AppComponent does not recover in case of a disruption event, you may experience unknown data loss and system downtime. If the alert criteria is set to **Red**, it indicates that the AppComponent is unrecoverable.

  Recommended action

  To ensure that your AppComponent is recoverable, review and implement the resiliency recommendations, and then run a new assessment. For more information about reviewing the resiliency recommendations, see the section called “Reviewing resiliency recommendations”.

For more information about using AWS Trusted Advisor, see the AWS Support User Guide.

Document history for the AWS Resilience Hub User Guide

The following table describes the documentation for this release of AWS Resilience Hub.

- **API version: latest**
- **Latest documentation update:** December 17, 2024

Change	Description	Date
Expanded Amazon EKS cluster permissions to include Ingress resources	Next generation Resilience Hub now reads Kubernetes Ingress resources to model the traffic path from Applicati on Load Balancers into cluster services. If you have included	October 14, 2026

	Amazon EKS clusters in your service, update the `ClusterRole` you granted for Next generation Resilience Hub to add `get` and `list` permissions on `ingresses` in the `networking.k8s.io` API group. If the permission is missing, the assessment still succeeds but the ingress-to-service traffic edges are not shown. For more information about updating the `ClusterRole` , see Amazon EKS permissions.	
Added the Amazon EBS pause action and updated minimum test duration for the Availability Zone: recovery test	The Availability Zone: recovery test now pauses I/O on Amazon EBS volumes in the impaired Availability Zone after stopped instances restart, and requires a test duration of at least 10 minutes. For more information, see Availability Zone: recovery.	October 1, 2026

Added organization policy sharing and the policy timeline	You can now share a resilience policy with the member accounts in your AWS organization, so service owners can apply the same policy to their own services. The account that owns a shared policy can review a timeline of the events that affected it, including when services across the organization started or stopped using it. For more information, see Sharing a policy across your organization.	September 16, 2026
Added Dependency insights	If you enable Dependency discovery, you can now generate Dependency insights, a generative AI-powered capability that analyzes discovered dependencies and highlights meaningful patterns, including cross-Region dependencies, new dependencies, third-party dependencies, and uneven usage patterns. For more information, see Dependency insights.	September 15, 2026

Expanded failure mode guidance and assertions	Expanded the failure mode guidance topic to describe how assertions work, how failure mode assessments use them, the assertion lifecycle across assessments, guidance for writing effective assertions, and how to manage assertions in the console and AWS CLI. Added the assertions-per-service quota. For more information, see Assertions.	September 11, 2026
Added EventBridge notification events	Next generation Resilience Hub now emits events to Amazon EventBridge for assessment lifecycle, failure mode finding resolution, and new dependency discovery. You can create EventBridge rules to automate responses to these events. For more information, see EventBridge notifications.	July 9, 2026

Documented dependency discovery status fields	Added documentation for the `eligibleResourceCount` and `message` fields in the `DependencyDiscoveryConfig` response structure. These fields allow you to monitor the status of dependency discovery through the `GetService` and `ListServices` API operations.	June 30, 2026
Added resilience testing IAM permissions	Added the trust and permissions policies for the IAM execution roles that AWS Fault Injection Service assumes to run resilience tests, for both single-account and multi-account tests.	June 15, 2026
Added resilience testing	Added documentation for resilience testing, which lets you validate service recovery using pre-configured tests built on AWS Fault Injection Service.	June 4, 2026
Initial release of Next generation Resilience Hub User Guide	Initial release of the Next generation Resilience Hub User Guide.	May 19, 2026

AWS Resilience Hub integrates already implemented Amazon CloudWatch alarms	AWS Resilience Hub now automatically detects and integrates already configured Amazon CloudWatch alarms into its resilience assessments, providing a more comprehensive view of your application's resilience posture. This new capability combines AWS Resilience Hub recommendations with your current monitoring setup to streamline alarm management and enhance assessment accuracy. For more information, see Managing alarms.	December 17, 2024

AWS Resilience Hub has enabled additional capabilities to provide simplified resilience testing with tailored AWS Fault Injection Service experiments	AWS Resilience Hub now supports an enhanced integration with AWS Fault Injection Service (AWS FIS) to offer tailored recommendations using AWS FIS actions and scenarios based on the specific application context to improve the resilience posture. Running the recommended experiments or your own tests will improve your resilience score, allowing you to track changes over time. For more information, see the following topics: • AWSResilienceHubAssessmentExecutionPolicy • Managing AWS Fault Injection Service experiments • AWS Resilience Hub – Resilience testing	December 17, 2024

AWS Resilience Hub introduces a summary view	AWS Resilience Hub's new summary view offers a high-level, visual representation of the applications' resilience through clear charts and graphs, allowing you to visualize the state of your application portfolio and efficiently manage and improve your applications' ability to withstand and recover from disruptions. In addition to the new summary view, you can export the data powering the summary view to create custom reports for stakeholder communication. For more information, see the section called “AWS Resilience Hub summary”.	November 21, 2024

AWS Resilience Hub introduces the **Resiliency widget** in the myApplications dashboard	The new **Resiliency widget** in the myApplications dashboard streamlines assessing and monitoring your applications' resilience posture. It enables you to quickly evaluate the resilience of applications defined in myApplications without having to manually replicate them in the AWS Resilience Hub. For more information, see the following topics: • the section called "AWS Resilience Hub and myApplications" • the section called "Managing resiliency assessments from Resiliency widget"	October 22, 2024

AWS Resilience Hub extends support for Amazon ElastiCache (Redis OSS) Serverless	AWS Resilience Hub now assesses applications that use Amazon ElastiCache (Redis OSS), including Amazon ElastiCache (Redis OSS) Serverless and Global Datastores, and provides enhanced resilience recommendations. These include guidelines for Region and multi-Region setups, as well as strategies for Multi-AZ deployments, resource grouping, and backup. Additionally, to provide improved control over the resilience posture of the applications, AWS Resilience Hub offers Amazon CloudWatch alarms that are tailored for Amazon ElastiCache (Redis OSS). For more information, see the following topics: • the section called "Managing Application Components" • the section called "Supported AWS Resilience Hub resources" • the section called "AWSResilienceHubAssessmentExecutionPolicy"	September 25, 2024

AWS Resilience Hub introduces grouping recommendations	AWS Resilience Hub introduces a new smart-grouping option to group resources into Application Components (AppComponents) while onboarding your applications. When you run resilience assessments on AWS Resilience Hub, it is important that your resources are accurately grouped into appropriate AppComponents to receive optimised and actionable recommendations. This option is ideal for complex or cross-Region applications to reduce the time taken to onboard your applications, and it complements the existing application onboarding workflow that is available today. For more information, see the following topics: • the section called "Managing Application Components" • the section called "AWS Resilience Hub resource grouping recommendations"	August 1, 2024

AWS Resilience Hub introduces a new assessment summary widget	AWS Resilience Hub introduces a new assessment summary widget that uses Amazon Bedrock generative AI capabilities to transform complex resilience data into highly actionable insights. These assessment summaries extract the critical findings, prioritize risks, and recommend steps to improve resilience. By focusing on the most impactful elements, you can understand the assessments much easier, which helps you with high-impact information that focuses on the most critical elements of your resilience posture. For more information, see the section called “Assessment Summary”.	August 1, 2024

AWS Resilience Hub extends support for Amazon DocumentDB	This AWS Resilience Hub policy allows you to grant `Describe` permissions to allow you to access resources and configurations on Amazon DocumentDB, Elastic Load Balancing, and AWS Lambda while running assessments. For more information about the AWS managed policy, see the section called “AWSResilienceHubAsssessmentExecutionPolicy”.	August 1, 2024

AWS Resilience Hub expands application resilience drift-detection capabilities	AWS Resilience Hub has expanded its drift detection capabilities by introducing a new type of drift detection – Application resource drift. This enhancement detects changes, such as addition or deletion of resources within the application's input sources. You can enable the AWS Resilience Hub scheduled assessment and drift notification services and be notified whenever a drift occurs. The latest resiliency assessment identifies the drifts and presents remediation actions to bring the application back into compliance with your resilience policy. For more information, see the following topics: • the section called “Drift detection” • the section called “Setup scheduled assessment and drift notification”	May 8, 2024

AWS Trusted Advisor enhancements	AWS Resilience Hub has expanded support for AWS Trusted Advisor by adding a check to identify unrecover able Application Components (AppComponents). For more information, see the section called “AWS Trusted Advisor”.	March 28, 2024
AWS Resilience Hub extends support for recommended alarms	AWS Resilience Hub has updated the `README.md` template file with values that allow you to create alarms recommended by AWS Resilience Hub within AWS (such as Amazon CloudWatch) or outside AWS. For more information, see the section called “Managing alarms”.	March 26, 2024

AWS Resilience Hub extends support for Amazon FSx for Windows File Server	AWS Resilience Hub extends assessment support for Amazon FSx for Windows File Server resources while assessing your applicati on's resiliency. For applicati ons using Amazon FSx for Windows File Server, AWS Resilience Hub provides a new set of resilience recommendations, covering Availability Zone (AZ) and Multi-AZ deployments, and backup plans, as well as data replication. AWS Resilience Hub supports Amazon FSx for Windows File Server, including filesyste m dependency on Microsoft Active Directory, for both in-Region and cross-Region deployments. For more information, see the following topics: • the section called "Supported AWS Resilience Hub resources" • the section called "AWSResilienceHubA sssessmentExecutionPolicy" • the section called "Grouping resources in an Application Component"	March 26, 2024

AWS Resilience Hub provides additional information about Resiliency score	AWS Resilience Hub has updated the Resiliency score user experience to help you easily navigate and understand the actions needed to improve the resilience posture of your applications. For more information, see the section called "Understanding resiliency scores".	November 9, 2023
AWS Resilience Hub extends support for applications that include Amazon Elastic Kubernetes Service (Amazon EKS) resources	AWS Resilience Hub extends the support for applications that include Amazon EKS resources to include new operational recommendations. While running an assessment that includes resources from Amazon EKS clusters, we will now recommend tests and alarms to be executed to help improve the resilience posture of the applications. For more information, see the section called "Managing AWS Fault Injection Service experiments".	November 9, 2023

AWS Resilience Hub provides additional information at the application level	AWS Resilience Hub provides additional information at the application level about estimated workload RTO and estimated workload RPO. This additional informati on indicates the maximum possible estimated workload RTO and estimated workload RPO of your application from the latest successfu l assessment. This value is the maximum estimated workload RTO and estimated workload RPO of all the disruption types. For more information, see the section called “Managing applications”.	October 30, 2023

AWS Resilience Hub extends assessment support for AWS Step Functions resources	AWS Resilience Hub extends assessment support for AWS Step Functions resources while assessing your applicati on's resiliency. AWS Resilienc e Hub analyzes the AWS Step Functions configuration including the state machine type (either Standard or Express workflows). In addition, AWS Resilienc e Hub will also provide recommendations that help you to meet the estimated workload Recovery Time Objectives (RTO) and estimated workload Recovery Point Objectives (RPO). To assess the applications including AWS Step Functions resources, you must set up the necessary permissions, either by using AWS managed policy or by manually adding the specific permission to allow AWS Resilience Hub to read the AWS Step Functions configuration. For more information about the associated permissio ns, see the section called "AWSResilienceHubA sssessmentExecutionPolicy".	October 30, 2023

AWS Resilience Hub allows Excluding Operational Recommendations	AWS Resilience Hub adds the ability for you to exclude operational recommendations including alarms, standard operating procedures (SOPs), and AWS Fault Injection Service (AWS FIS) tests. While running an assessment on AWS Resilience Hub, you are provided estimated recovery times and recommendations on ways to increase the resilience of the applicati on that was assessed. Using the exclude recommend ations workflow, you will now have the ability to exclude recommended alarms, SOPs, and AWS FIS tests that are not relevant for them. The exclude workflow is beneficia l if you are using a platform outside of one suggested, or have already implement ed the recommendation in alternative method. For more information, see the following topics: • the section called “Includin g or excluding operational recommendations” • the section called “Limiting permissions to include or	August 9, 2023

	exclude AWS Resilience Hub recommendations”	
Improving permissions design for AWS Resilience Hub	AWS Resilience Hub introduces a new permission design to provide flexibility while configuring AWS Identity and Access Management (IAM) roles for AWS Resilience Hub. It also consolidates permissions into a single role, with the ability to create custom role names that are meaningful to you and your teams. A new managed policy in AWS Resilience Hub will allow you to have the appropriate permissions for the supported services. If you are comfortable with the current method of setting permissions, we will continue to support the manual configuration. For more information about the AWS managed policy, see the section called “AWSResilienceHubAsssessmentExecutionPolicy”.	August 2, 2023

Application Resilience Drift Detection with AWS Resilience Hub	AWS Resilience Hub allows you to proactively detect and understand the necessary actions to resolve application resilience. Enabling Amazon Simple Notification Service (Amazon SNS) to receive notifications when the estimated workload recovery time objective (RTO) or estimated workload recovery point objective (RPO) has moved from meeting the target to no longer satisfying your organization's business objectives. Moving from reactively finding resilience issues while manually running an assessment to proactively being notified through Amazon SNS topics will allow you to anticipate potential disruptions earlier, and provide additional confidence that recovery objectives will be achieved. For more information, see the following topics: • the section called "Setup scheduled assessment and drift notification" • the section called "Editing application resources"	August 2, 2023

AWS Resilience Hub improves support for Amazon Relational Database Service and Amazon Aurora	AWS Resilience Hub extends assessment support for Amazon Relational Database Service proxy, and headless and Amazon Aurora DB database configurations. In addition, while assessing applications that include Amazon RDS, we will now distinguish between different database engines to provide more precise estimated workload recovery time objectives (RTOs). AWS Resilience Hub will also provide additional actions to implement resilience best practices within your AWS environment. The best practices can include performance insights with DevOps Guru for Amazon RDS, enhanced monitoring, and blue/green deployment automation on supported database engines. To learn more about the permissions required for AWS Resilience Hub to include resources from all the supported services in your assessment, see the section called “AWSResilienceHubAssessmentExecutionPolicy”.	August 2, 2023

AWS Resilience Hub extends support for Amazon Elastic Block Store snapshots	AWS Resilience Hub extends assessment support for Amazon Elastic Block Store (Amazon EBS) to recognize Amazon EBS snapshots, which are taken within the same Amazon EBS Region using direct APIs. The extended support is in addition to current support for customers using Amazon Data Lifecycle Manager (Amazon Data Lifecycle Manager) or AWS Backup. For more information, see Amazon Elastic Block Store (Amazon EBS).	August 2, 2023

Amazon Elastic Compute Cloud enhancements	AWS Resilience Hub has expanded support for Amazon Elastic Compute Cloud (Amazon EC2). For Applications of different sizes, AWS allows its customers using Amazon EC2 to select the configuration that is appropriate for their use case. AWS Resilience Hub supports assessment on the following Amazon EC2 configurations: • On-demand instances. • Instances backup by AWS Backup and AWS Elastic Disaster Recovery. • Support for auto-scal ing groups with Amazon Application Recovery Controller (ARC) (ARC) Going forward, assessment support will extend to include spot instances, dedicated hosts, dedicated instances, placement groups, and fleets. For more information, see the section called “AWS Resilienc e Hub access permissions reference”.	June 27, 2023

AWS managed policy updates	Added a new policy that provides access to other AWS services for executing assessments. For more information, see the section called “AWSResilienceHubAsssessmentExecutionPolicy”.	June 26, 2023
New Amazon DynamoDB operational recommendation alarms	For applications using Amazon DynamoDB, AWS Resilience Hub now provides a new set of alarms that alert you to resilience risks for on-demand and provisioned capacity modes and global tables. To access the new alarms, you may need to update the AWS Identity and Access Management (IAM) policy of the role you are using. For more information, see the section called “AWS Resilience Hub access permissions reference”.	May 2, 2023

AWS Trusted Advisor enhancements	AWS Resilience Hub has expanded support for AWS Trusted Advisor and the applications using Amazon DynamoDB. When you use AWS Trusted Advisor with AWS Resilience Hub, you can now receive a notification when an application has not been assessed in the previous 30 days. This notification prompts you to reassess the application to understand if there are any changes that would impact its resiliency. For more information about **AWS Resilience Hub assessment age** check, see the section called "AWS Trusted Advisor".	May 2, 2023

Additional support for Amazon Simple Storage Service	In addition to the current support of Amazon Simple Storage Service (Amazon S3) Cross-Region Replication (Amazon S3 CRR)/ Amazon S3 Same-Region Replication (SRR), versioning, and AWS Backup, AWS Resilience Hub will now assess Amazon S3 for multi-Region access point, Amazon S3 Replication Time Control (Amazon S3 RTC), and AWS Backup point-in-time recovery (PITR) configuration. For more information, see the following topics: • the section called “AWS Resilience Hub access permissions reference” • Managing your Amazon S3 storage	March 21, 2023

Additional support for Amazon Elastic Kubernetes Service	AWS Resilience Hub has added Amazon EKS cluster as a supported resource for defining, validating, and tracking application resiliency. Customers can add Amazon EKS clusters to new or existing applicati ons, and receive assessmen ts and recommendations for improving resiliency. Customers can add applicati on resources using AWS CloudFormation, Terraform , AWS Resource Groups, and myApplications. Additiona lly, customers can add one or more Amazon EKS clusters directly in one or more Regions with one or more namespaces in each cluster. This allows AWS Resilience Hub to provide single and cross-Region assessments and recommendations. In addition to examining deployments, Replicas, ReplicationControl lers, and Pods, AWS Resilienc e Hub will analyze the overall cluster resiliency. AWS Resilience Hub supports stateless Amazon EKS cluster workloads. The new capabilit ies are available in all the AWS Regions where AWS Resilience Hub is supported.	March 21, 2023

	For more information, see the following topics: • the section called “Manage your application resources” • the section called “Add EKS clusters” • the section called “AWS Resilience Hub access permissions reference” • AWS Regional Services	
Additional support for Amazon Elastic File System	In addition to the current support for Amazon Elastic File System (Amazon EFS) backup, AWS Resilience Hub will now assess Amazon EFS for Amazon EFS replication and AZ configuration. For more information, see the following topics: • the section called “Supported AWS Resilience Hub resources” • What is Amazon Elastic File System?	March 21, 2023

Support for application input sources	AWS Resilience Hub now provides transparency about your application sources. It helps you to add, delete, and reimport input sources of your application, and publish a new application version. For more information, see the section called "Editing application resources".	February 21, 2023
Support for application configuration parameters	AWS Resilience Hub now provides an input mechanism to gather additional informati on about the resources associated with your applicati ons. With this information, AWS Resilience Hub will gain a deeper understanding of your resources and provide better resiliency recommend ations. For more information, see the following topics: • the section called "Application configuration parameters" • the section called "Configur e the application configura tion parameters" • the section called "Updating application configuration parameters"	February 21, 2023

Additional support for Amazon Elastic Block Store	In addition to the current support of Amazon Elastic Block Store (Amazon EBS) volumes, AWS Resilience Hub will now asses Amazon EBS snapshots by Amazon Data Lifecycle Manager and Amazon EBS fast snapshot restore (FSR). For more information, see the following topics: • the section called "AWS Resilience Hub access permissions reference" • Amazon Elastic Block Store (Amazon EBS)	February 21, 2023

Integration with AWS Trusted Advisor	AWS Trusted Advisor users will be able to view applications associated with their account that have been assessed by AWS Resilience Hub. AWS Trusted Advisor shows the latest resilience score and provides a status that indicates if the targeted resilience policy (RTO and RPO) has been met or not. Each time an assessment is run, AWS Resilience Hub updates AWS Trusted Advisor with the latest results. AWS Trusted Advisor is a service that continuously analyzes your AWS accounts and provides recommendations to help you to follow AWS best practices and AWS Well-Architected guidelines. For more information, see the section called "AWS Trusted Advisor".	November 18, 2022

Support for Amazon Simple Notification Service (Amazon SNS)	AWS Resilience Hub now assesses applications using Amazon SNS by analyzing Amazon SNS configura tion, including subscribers, and provides recommend ations to meet the organizat ion's estimated workload recovery objectives (estimated workload RTO and estimated workload RPO) for the applications. Amazon SNS is a managed service that delivers message from publishers (producers) to subscribers (consumers). For more information, see the following topics: • the section called "Supported AWS Resilience Hub resources" • the section called "Identity and Access Management" • the section called "Grouping resources in an Application Component"	November 16, 2022

Additional Support for Amazon Application Recovery Controller (ARC) (Amazon ARC)	AWS Resilience Hub now assesses Amazon ARC for Elastic Load Balancing and Amazon Relational Database Service (Amazon RDS), which includes advising when Amazon ARC would be beneficial. Extending AWS Resilience Hub, Amazon ARC assessment support beyond AWS Auto Scaling Group (AWS ASG) and Amazon DynamoDB. Amazon ARC provides high availability for your application, allowing you to quickly failover your entire application to a failover Region. For more information, see the following topics: • the section called "Supported AWS Resilience Hub resources" • the section called "Identity and Access Management"	November 16, 2022

Additional Support for AWS Backup	AWS Resilience Hub now assesses Amazon ARC for Elastic Load Balancing and Amazon Relational Database Service (Amazon RDS), which includes advising when Amazon ARC would be beneficial. Extending AWS Resilience Hub, Amazon ARC assessment support beyond AWS Auto Scaling Group (AWS ASG) and Amazon DynamoDB. Amazon ARC provides high availability for your application, allowing you to quickly failover your entire application to a failover Region. For more information, see the following topics: • the section called "Supported AWS Resilience Hub resources" • the section called "Identity and Access Management"	November 16, 2022
Updated content: Added new Application Component resources	Added Route53 and AWS Backup to the list of supported Application Component resources in the AppComponent grouping section.	July 1, 2022
New content: Application compliance status concept	Added the Changes detected status type.	June 2, 2022

Introducing AWS Resilience Hub	AWS Resilience Hub is now available. This guide describes how to use AWS Resilience Hub to analyze your infrastructure, get recommendations to improve the resiliency of your AWS apps, review resiliency scores, and more.	November 10, 2021

Next generation Resilience Hub

Next generation Resilience Hub is a resilience management service that uses GenAI-powered failure mode assessments, automatic dependency discovery, resilience testing, and composable resilience policies to help you proactively assess, test, and improve the resilience posture of your AWS applications.

Topics

- What is Next generation Resilience Hub?
- Setting up Next generation Resilience Hub
- Getting started with Next generation Resilience Hub
- Systems, user journeys, and services
- Resilience policies
- Assessment
- Resilience testing
- Dashboard and reporting
- Working with AWS DevOps Agent in the next generation of Resilience Hub
- AWS Organizations integration
- Migrating from AWS Resilience Hub
- Security in Next generation Resilience Hub
- Monitoring Next generation Resilience Hub
- Quotas and limits
- Troubleshooting Next generation Resilience Hub
- API reference
- Document history for the Next generation Resilience Hub User Guide

What is Next generation Resilience Hub?

Next generation Resilience Hub is a resilience management service that helps you define resilience goals, model your applications, discover dependencies, test resilience, and assess your resilience posture against those goals. Next generation Resilience Hub uses GenAI-powered failure mode assessments based on the AWS Well-Architected Framework to identify potential weaknesses

and provide actionable recommendations. AWS recommended resilience tests validate how your services respond to and recover from specific failure scenarios aligned to fault boundaries and dependencies.

Key capabilities at a glance

Next generation Resilience Hub provides the following key capabilities:

- **Application modeling** – Model your applications using a hierarchy that matches how your organization thinks about them. Systems represent your business applications, user journeys describe critical end-user paths within a system, and services are the building blocks comprising AWS resources that support user journeys. You define where to find your AWS resources (AWS CloudFormation stacks, Terraform state files, resource tags, or Amazon EKS clusters), and Resilience Hub automatically discovers and maps them into a topology showing how resources connect.
- **Dependency discovery** – Continuously discovers and maps dependencies for your services, including AWS services, internal endpoints, and third-party endpoints.
- **Failure mode assessments (GenAI-powered)** – Analyzes your services against resilience policies and AWS Well-Architected best practices to identify potential failure modes and recommend improvements.
- **Resilience tests** – Tests your services to verify recovery objectives are met and dependencies behave as expected, such as Availability Zone impairment, Regional impairment, and dependency failure.
- **Resilience policies** – Modular, composable requirements for availability SLO, Multi-Region disaster recovery, Multi-AZ disaster recovery, and data recovery objective.
- **AWS Organizations integration** – Centralized governance across multiple accounts with delegated administrator support and organization-wide resilience policies.

For enterprise-scale deployments, Next generation Resilience Hub integrates with AWS Organizations to provide centralized resilience management across your entire organization from a single delegated administrator account. You get organization-wide visibility into resilience posture and aggregated dashboards without logging in to individual accounts.

Supported Regions and availability

Next generation Resilience Hub is available in the following AWS Regions.

Region name	Region code
US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (Oregon)	us-west-2
Canada (Central)	ca-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Frankfurt)	eu-central-1
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Asia Pacific (Seoul)	ap-northeast-2
South America (São Paulo)	sa-east-1

For the latest information about available AWS Regions, see Next generation Resilience Hub endpoints and quotas in the AWS General Reference.

Pricing overview

Next generation Resilience Hub uses a service-based pricing model. The following table summarizes the pricing components.

Component	Price
Service fee (includes up to 150 resources and 2 failure mode assessments per month)	$15 per service per month
Additional resources above 150 and each assessment after 2 included	$0.10 per resource per failure mode assessment
Dependency discovery (optional add-on)	$10 per service per month
Resilience tests (optional)	$0.10 per Fault Injection Service (FIS) action-minute

The service fee includes two failure mode assessments per month for services with up to 150 resources. For services with more than 150 resources, each included failure mode assessment incurs an additional charge of $0.10 per resource above the 150-resource threshold. For additional assessments beyond the two included, the cost is $0.10 per assessable resource with a minimum of 50 resources.

Dependency discovery is an optional add-on that provides continuous, automated identification of all dependencies your service calls.

Resilience tests are powered by AWS Fault Injection Service (AWS FIS). You are charged $0.10 per action-minute plus $0.10 per action-minute for each additional target account. For detailed pricing information, see AWS FIS pricing. Resilience test reports generated by the next generation of Resilience Hub are included at no additional cost.

For detailed pricing information, see Next generation Resilience Hub pricing.

Related services

The following AWS services integrate with or complement the next generation of Resilience Hub:

- **AWS Fault Injection Service (AWS FIS)** – Resilience tests use AWS FIS to inject controlled faults into your service, to help you validate it operates and recovers as expected. For more information, see Resilience testing.
- **AWS Application Recovery Controller (ARC)** – Provides readiness checks, routing controls, and zonal shift capabilities that complement the next generation of Resilience Hub.

- **AWS Well-Architected Framework** – Failure mode assessments leverage Well-Architected best practices to evaluate your service architecture.
- **AWS Organizations** – Enables multi-account resilience governance through the next generation of Resilience Hub from a single delegated administrator account.
- **Amazon EventBridge** – Use Amazon EventBridge to receive notification events from the next generation of Resilience Hub for assessment completion, finding resolution, and dependency discovery. You can use these events to build event-driven automation workflows.

Setting up Next generation Resilience Hub

Before you can use the next generation of Resilience Hub, you must complete the setup steps described in this section.

Topics

- Prerequisites
- Required IAM permissions and roles
- AWS managed policies for Next generation Resilience Hub
- Configuring AWS Organizations integration (optional)
- Multi-account setup (without AWS Organizations)
- Customer-managed keys (optional)

Prerequisites

Before you begin, ensure that you have the following:

- An active AWS account.
- IAM permissions to create roles and policies in your account.
- (Optional) AWS Organizations configured if you plan to use multi-account governance.
- (Optional) AWS resources deployed via AWS CloudFormation, Terraform, resource tags, or Amazon Elastic Kubernetes Service that you want to assess.
- (Optional) An IAM execution role for resilience testing. For details, see IAM execution roles for resilience testing.

Required IAM permissions and roles

AWS managed policy

You can attach the `AWSResilienceHubV2AssessmentExecutionPolicy` to your IAM identities. While running an assessment, this policy grants read-only access permissions to other AWS services for resilience discovery, assessment, and management. For details about the permissions included in this policy, see AWSResilienceHubV2AssessmentExecutionPolicy.

> **Note**
>
> The `AWSResilienceHubV2AssessmentExecutionPolicy` replaces the previous `AWSResilienceHubAsssessmentExecutionPolicy` for use with the next generation of Resilience Hub.

If you use resilience testing, also attach the `AWSResilienceHubResilienceTestingPolicy` managed policy. This policy grants Resilience Hub the AWS Fault Injection Service (AWS FIS) permissions needed to start and manage experiments on your behalf. For details about the permissions included in this policy, see AWSResilienceHubResilienceTestingPolicy.

> **Note**
>
> If the managed policy is not available in your account, create an inline policy with the same permissions. For the policy contents, see AWSResilienceHubResilienceTestingPolicy.

IAM role for assessment and resilience testing

To run an assessment or a resilience test, the next generation of Resilience Hub must assume an IAM role with the required permissions. This role discovers and reads the configuration of your AWS resources, and — when resilience testing is enabled — starts and manages AWS FIS experiments on your behalf.

There are two ways to create or configure this role:

- **Create a new service role (recommended)** – When you create or edit a service in the Next generation Resilience Hub console, you can choose **Create new role** under **Permission model**.

The console automatically creates an IAM role with the correct trust policy and attaches the `AWSResilienceHubV2AssessmentExecutionPolicy` managed policy.

- **Use an existing service role** – If you already have a role configured, or if you prefer to create roles outside of the console, choose **Use an existing service role**. Then choose the role from the list and make sure that the role has the required trust policy and permissions described below.

Creating the role manually

To create the role manually, open the IAM console. Choose **Custom trust policy** and use a trust policy like this:

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

For permissions, attach the `AWSResilienceHubV2AssessmentExecutionPolicy` managed policy. If you use resilience testing, also attach the `AWSResilienceHubResilienceTestingPolicy` managed policy.

IAM Service-Linked Role

Next generation Resilience Hub automatically creates a Service-Linked Role with the `AWSResilienceHubServiceRolePolicy` managed policy.

Terraform state file access permissions

If you are including Terraform state files into your Next generation Resilience Hub service, provide permissions to read the Terraform files from your Amazon S3 bucket with a policy like this:

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::s3-bucket-name/path-to-state-file"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::s3-bucket-name"
    }
  ]
}
```

Amazon EKS Permissions

If you are including Amazon EKS clusters into your Next generation Resilience Hub service, follow the following 3-step process to provide Next generation Resilience Hub permissions to read configuration data for your Amazon EKS clusters using Kubernetes role-based access control (RBAC).

Each step also has a part labeled **(Optional) Resilience testing**. Complete these parts only if you run resilience tests that inject faults into Amazon EKS pods. Three test templates do this when they block dependencies: **Dependency validation**, **Multi-Region: isolation**, and **Multi-Region: recovery**. The Amazon EKS pod actions in these templates need setup inside your cluster that you can't configure from the next generation of Resilience Hub. This setup includes a Kubernetes service account with fault-injection permissions. It also includes cluster access for the test execution role. If your service has no Amazon EKS pods, or you don't use resilience testing, skip these parts.

The resilience testing setup is separate from, and in addition to, the discovery setup. Discovery grants a read-only cluster role to a Kubernetes group that your service role maps to. Resilience testing grants fault-injection permissions to a Kubernetes service account and to a Kubernetes user that your test execution role maps to. A cluster that you both assess and test needs both.

Important

Because the resilience testing setup lives inside your cluster, the next generation of Resilience Hub can't validate it when you create a test or start a test run. Setup problems surface only when the test runs.
If the service account is missing, or the execution role lacks cluster access, the test run still starts. The Amazon EKS pod action then fails during fault injection with a Kubernetes authorization error.

(Optional) Resilience testing: Prerequisites

Before you complete the resilience testing parts of the following steps, make sure that:

- Your cluster runs Amazon EKS version 1.30 or later.
- Your target pods run on Amazon EC2 nodes. The packet loss action that these tests use isn't supported on AWS Fargate.
- The `securityContext` of your target pods sets `readOnlyRootFilesystem: false`. Amazon EKS pod actions fail without it.
- You have `kubectl` configured for the cluster, and permissions to create RBAC resources in the namespaces you want to test.

Note

In addition to the cluster setup, the test execution role needs `eks:DescribeCluster` on the cluster, along with the other permissions for the test template you're using. The permissions policies in IAM execution roles for resilience testing already include these for the templates that support Amazon EKS pods.

Step 1: Apply the following to your Amazon EKS cluster

This grants Next generation Resilience Hub read-only access to the Kubernetes resources it needs across all namespaces:

```
cat << EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
```

```
kind: ClusterRole
metadata:
  name: resilience-hub-eks-access-cluster-role
rules:
- apiGroups:
    - ""
  resources:
    - pods
    - replicationcontrollers
    - nodes
    - services
  verbs:
    - get
    - list
- apiGroups:
    - apps
  resources:
    - deployments
    - replicasets
  verbs:
    - get
    - list
- apiGroups:
    - networking.k8s.io
  resources:
    - ingresses
  verbs:
    - get
    - list
- apiGroups:
    - policy
  resources:
    - poddisruptionbudgets
  verbs:
    - get
    - list
- apiGroups:
    - autoscaling.k8s.io
  resources:
    - verticalpodautoscalers
  verbs:
    - get
    - list
- apiGroups:
```

```
    - autoscaling
  resources:
    - horizontalpodautoscalers
  verbs:
    - get
    - list
- apiGroups:
    - karpenter.sh
  resources:
    - provisioners
    - nodepools
  verbs:
    - get
    - list
- apiGroups:
    - karpenter.k8s.aws
  resources:
    - awsnodetemplates
    - ec2nodeclasses
  verbs:
    - get
    - list
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: resilience-hub-eks-access-cluster-role-binding
subjects:
  - kind: Group
    name: resilience-hub-eks-access-group
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: resilience-hub-eks-access-cluster-role
  apiGroup: rbac.authorization.k8s.io
---
EOF
```

(Optional) Resilience testing: Create the Kubernetes service account

Resilience tests run Amazon EKS pod actions as a Kubernetes service account named exactly `resilience-testing-service-account`. You can't change this name—the next generation of Resilience Hub uses it for every test run.

> **Important**
>
> Service accounts are scoped to a namespace. Create `resilience-testing-service-account` in every namespace that contains pods you want to test. If a test targets pods in a namespace where the service account doesn't exist, the Amazon EKS pod action fails for those pods.

The following manifest creates the service account, a role granting the permissions that AWS FIS needs to inject faults, and a role binding that ties them together. Replace *`namespace`* with your target namespace.

```
kind: ServiceAccount
apiVersion: v1
metadata:
  namespace: namespace
  name: resilience-testing-service-account

---
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: namespace
  name: resilience-testing-role
rules:
- apiGroups: [""]
  resources: ["configmaps"]
  verbs: ["get", "create", "patch", "delete"]
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["create", "list", "get", "delete", "deletecollection"]
- apiGroups: [""]
  resources: ["pods/ephemeralcontainers"]
  verbs: ["update"]
- apiGroups: [""]
  resources: ["pods/exec"]
```

```
  verbs: ["create"]
- apiGroups: ["apps"]
  resources: ["deployments"]
  verbs: ["get"]

---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: resilience-testing-role-binding
  namespace: namespace
subjects:
- kind: ServiceAccount
  name: resilience-testing-service-account
  namespace: namespace
- apiGroup: rbac.authorization.k8s.io
  kind: User
  name: resilience-testing-user
roleRef:
  kind: Role
  name: resilience-testing-role
  apiGroup: rbac.authorization.k8s.io
```

Save the manifest as `rbac.yaml` and apply it:

```
kubectl apply -f rbac.yaml
```

The role binding has two subjects. The service account is what the fault-injection pod runs as. The Kubernetes user is the identity that your test execution role maps to in Step 2. You choose the user name—`resilience-testing-user` in this example—but it must match the user name you use in Step 2.

Repeat this part for each namespace you want to test. The role and role binding names can be the same in every namespace. Only the service account name is fixed by the next generation of Resilience Hub.

Step 2: Map the IAM role to the Kubernetes group

Map the IAM role you created to the `resilience-hub-eks-access-group` Kubernetes group. You can use either Amazon EKS access entries (recommended) or the `aws-auth` ConfigMap.

Option A: Using EKS access entries (recommended)

EKS access entries are the preferred method for managing cluster authentication. Your cluster must use `API` or `API_AND_CONFIG_MAP` authentication mode.

```
aws eks create-access-entry \
  --cluster-name cluster-name \
  --principal-arn arn:aws:iam::ACCOUNT-ID:role/ResilienceHubRole \
  --type STANDARD \
  --kubernetes-groups '["resilience-hub-eks-access-group"]'
```

Option B: Using aws-auth ConfigMap

If your cluster uses `CONFIG_MAP` or `API_AND_CONFIG_MAP` authentication mode, you can edit the aws-auth ConfigMap instead:

Using eksctl:

```
eksctl create iamidentitymapping \
  --cluster cluster-name \
  --region region \
  --arn arn:aws:iam::ACCOUNT-ID:role/ResilienceHubRole \
  --group resilience-hub-eks-access-group \
  --username AwsResilienceHubAssessmentEKSAccessRole
```

Or manually edit the ConfigMap:

```
kubectl edit -n kube-system configmap/aws-auth
```

Add this under `mapRoles` in the data section:

```
- groups:
    - resilience-hub-eks-access-group
  rolearn: arn:aws:iam::ACCOUNT-ID:role/ResilienceHubRole
  username: AwsResilienceHubAssessmentEKSAccessRole
```

(Optional) Resilience testing: Grant the execution role access to the cluster

For resilience testing, you map a different IAM role and a different Kubernetes identity than you do for discovery. Map your *test execution role*—not the service role you mapped above—to the Kubernetes *user* in the role binding from Step 1, so that AWS FIS can create the fault-injection pod. Don't add the execution role to the `resilience-hub-eks-access-group` group. As with discovery, use an Amazon EKS access entry (recommended) or the `aws-auth` ConfigMap. The same authentication mode requirements apply: access entries need `API` or `API_AND_CONFIG_MAP`, and the `aws-auth` ConfigMap needs `CONFIG_MAP` or `API_AND_CONFIG_MAP`.

> **Important**
>
> Map the execution role from the account that contains the Amazon EKS cluster. For a multi-account test, this is the *target account* role, not the orchestrator account role. Mapping the orchestrator role is a common mistake. It leaves the test failing with a Kubernetes authorization error. The role that actually reaches the cluster is the one AWS FIS assumes in the target account. For more information about multi-account execution roles, see Multi-account execution roles.

Using an access entry (Option A):

```
aws eks create-access-entry \
  --cluster-name cluster-name \
  --principal-arn arn:aws:iam::cluster-account-id:role/execution-role-name \
  --username resilience-testing-user
```

Using the aws-auth ConfigMap with eksctl (Option B):

```
eksctl create iamidentitymapping \
  --cluster cluster-name \
  --region region \
  --arn arn:aws:iam::cluster-account-id:role/execution-role-name \
  --username resilience-testing-user
```

Or, if you edit the ConfigMap manually, add this under `mapRoles`:

```
- rolearn: arn:aws:iam::cluster-account-id:role/execution-role-name
  username: resilience-testing-user
```

> **Note**
>
> Entries in the `aws-auth` ConfigMap don't support a path component in the role ARN. Use `arn:aws:iam::cluster-account-id:role/execution-role-name`, not `arn:aws:iam::cluster-account-id:role/service-role/execution-role-name`.

Step 3: Verify

Confirm the RBAC resources exist and the role mapping is in place:

```
kubectl get clusterrole resilience-hub-eks-access-cluster-role
kubectl describe clusterrolebinding resilience-hub-eks-access-cluster-role-binding
```

If using access entries (Option A):

```
aws eks describe-access-entry \
  --cluster-name cluster-name \
  --principal-arn arn:aws:iam::ACCOUNT-ID:role/ResilienceHubRole
```

If using aws-auth ConfigMap (Option B):

```
kubectl get configmap aws-auth -n kube-system -o yaml | grep -A 3 "ResilienceHubRole"
```

(Optional) Resilience testing: Verify the service account and execution role mapping

Confirm the service account and role binding exist in each namespace you plan to test:

```
kubectl get serviceaccount resilience-testing-service-account -n namespace
kubectl describe rolebinding resilience-testing-role-binding -n namespace
```

If you created an access entry for the execution role (Option A), confirm the mapping:

```
aws eks describe-access-entry \
  --cluster-name cluster-name \
  --principal-arn arn:aws:iam::cluster-account-id:role/execution-role-name
```

If you used the aws-auth ConfigMap (Option B):

```
kubectl get configmap aws-auth -n kube-system -o yaml | grep -A 3 "execution-role-name"
```

Important

Check that the user name in the access entry or ConfigMap entry matches a `User` subject in the role binding. Both halves of the setup can look correct on their own: the cluster authenticates the execution role, and the RBAC resources exist. If the two names differ, the Amazon EKS pod action fails at injection time with a Kubernetes authorization error. A mismatch here is the most common cause of Amazon EKS pod actions failing.

AWS managed policies for Next generation Resilience Hub

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

Topics

- AWSResilienceHubV2AssessmentExecutionPolicy
- AWSResilienceHubResilienceTestingPolicy
- AWSResilienceHubServiceRolePolicy
- Next generation Resilience Hub updates to AWS managed policies

AWSResilienceHubV2AssessmentExecutionPolicy

You can attach the `AWSResilienceHubV2AssessmentExecutionPolicy` to your IAM identities. While running an assessment, this policy grants read-only access permissions to other AWS services for resilience discovery, assessment, and management.

Permission details

This policy grants wildcarded read-only permissions that might include sensitive information in the output.

This policy includes the following permissions:

- Amazon CloudWatch (CloudWatch) – Provides `Describe`, `Get`, and `List` permissions for CloudWatch resources that are associated with your AWS account.
- AWS CloudFormation – Provides `Describe`, `Get`, and `List` permissions for AWS CloudFormation resources that are associated with your AWS account.
- Amazon Elastic Compute Cloud (Amazon EC2) – Provides specific `Describe` permissions for Amazon EC2 resources that are associated with your AWS account.
- Amazon Elastic Container Service (Amazon ECS) – Provides `Describe` and `List` permissions for Amazon ECS resources that are associated with your AWS account.
- Amazon Elastic Kubernetes Service (Amazon EKS) – Provides `Describe` and `List` permissions for Amazon EKS resources that are associated with your AWS account.
- Amazon Elastic Container Registry (Amazon ECR) – Provides `Describe` permissions for Amazon ECR resources that are associated with your AWS account.

- Amazon Elastic File System (Amazon EFS) – Provides `Describe` permissions for Amazon EFS resources that are associated with your AWS account.
- Amazon ElastiCache (ElastiCache) – Provides `Describe` permissions for ElastiCache resources that are associated with your AWS account.
- Elastic Load Balancing – Provides `Describe` permissions for Elastic Load Balancing resources that are associated with your AWS account.
- Amazon DynamoDB (DynamoDB) – Provides `Describe` and `List` permissions for DynamoDB resources that are associated with your AWS account.
- Amazon RDS – Provides `Describe` permissions for Amazon RDS resources that are associated with your AWS account.
- Amazon DocumentDB – Provides `Describe` and `List` permissions for Amazon DocumentDB resources that are associated with your AWS account.
- AWS Lambda (Lambda) – Provides specific `Get` and `List` permissions for Lambda resources that are associated with your AWS account.
- AWS Step Functions – Provides `Describe` and `List` permissions for AWS Step Functions resources that are associated with your AWS account.
- IAM – Provides specific `Get` and `List` permissions for IAM resources that are associated with your AWS account.
- Amazon Simple Notification Service (Amazon SNS) – Provides `Get` and `List` permissions for Amazon SNS resources that are associated with your AWS account.
- Amazon Simple Queue Service (Amazon SQS) – Provides `Get` and `List` permissions for Amazon SQS resources that are associated with your AWS account.
- Amazon Simple Storage Service (Amazon S3) – Provides `Get` and `List` permissions for Amazon S3 resources that are associated with your AWS account. The Amazon S3 permissions in the `AWSResilienceHubS3AccessStatement` are restricted to resources in the same account by using the `aws:ResourceAccount` condition key.
- Amazon Route 53 (Route 53) – Provides `Get` and `List` permissions for Route 53 resources, including Route 53 Application Recovery Controller resources, that are associated with your AWS account.
- Amazon EC2 Systems Manager (SSM) – Provides `Describe` and `Get` permissions for SSM resources that are associated with your AWS account.
- Amazon EC2 Auto Scaling – Provides `Describe` permissions for Amazon EC2 Auto Scaling resources that are associated with your AWS account.

- AWS Backup – Provides `Describe`, `Get`, and `List` permissions for AWS Backup resources that are associated with your AWS account.
- AWS Elastic Disaster Recovery (Elastic Disaster Recovery) – Provides `Describe` permissions for Elastic Disaster Recovery resources that are associated with your AWS account.
- AWS Fault Injection Service (AWS FIS) – Provides `Get` and `List` permissions for AWS FIS experiments and experiment templates that are associated with your AWS account.
- Amazon FSx for Windows File Server (Amazon FSx) – Provides `Describe` permissions for Amazon FSx resources that are associated with your AWS account.
- Amazon Data Lifecycle Manager – Provides `Get` permissions for Amazon Data Lifecycle Manager resources that are associated with your AWS account.
- AWS DataSync – Provides `Describe` and `List` permissions for AWS DataSync resources that are associated with your AWS account.
- AWS Resource Groups (Resource Groups) – Provides `Get` and `List` permissions for Resource Groups resources that are associated with your AWS account.
- AWS Service Catalog (Service Catalog) – Provides `Get` and `List` permissions for Service Catalog resources that are associated with your AWS account.
- Amazon API Gateway – Provides `GET` permissions scoped to specific resource ARN patterns for REST APIs, HTTP APIs, usage plans, and domain names.
- Amazon Kinesis – Provides `Describe` and `List` permissions for Kinesis resources that are associated with your AWS account.
- Amazon Kinesis Data Firehose – Provides `Describe` and `List` permissions for Kinesis Data Firehose resources that are associated with your AWS account.
- Amazon EventBridge – Provides `Describe` and `List` permissions for EventBridge resources that are associated with your AWS account.
- Amazon MSK – Provides `Describe`, `Get`, and `List` permissions for Amazon MSK and MSK Connect resources that are associated with your AWS account.
- Amazon MemoryDB – Provides `Describe` permissions for MemoryDB resources that are associated with your AWS account.
- Amazon Redshift – Provides `Describe` permissions for Redshift resources that are associated with your AWS account.
- AWS Global Accelerator – Provides `Describe` and `List` permissions for Global Accelerator resources that are associated with your AWS account.

- AWS Network Firewall – Provides `Describe` and `List` permissions for Network Firewall resources that are associated with your AWS account.
- AWS Shield – Provides `Describe` and `List` permissions for Shield resources that are associated with your AWS account.
- AWS WAF V2 – Provides `Get` and `List` permissions for WAF V2 resources that are associated with your AWS account.
- AWS Resource Access Manager – Provides `Get` and `List` permissions for RAM resources that are associated with your AWS account.
- Amazon VPC Lattice – Provides `Get` and `List` permissions for VPC Lattice resources that are associated with your AWS account.
- AWS Config – Provides `Describe` and `List` permissions for Config resources that are associated with your AWS account.
- Amazon CloudFront – Provides `Get` and `List` permissions for CloudFront resources that are associated with your AWS account.
- AWS Secrets Manager – Provides `Describe` and `List` permissions for Secrets Manager resources that are associated with your AWS account.
- AWS Directory Service – Provides `Describe` permissions for Directory Service resources that are associated with your AWS account.
- Amazon DSQL – Provides `Get` and `List` permissions for DSQL resources that are associated with your AWS account.
- Amazon QLDB – Provides `Describe` and `List` permissions for QLDB resources that are associated with your AWS account.
- AWS Certificate Manager – Provides `Describe`, `Get`, and `List` permissions for ACM resources that are associated with your AWS account.
- Application Auto Scaling – Provides `Describe` permissions for Application Auto Scaling resources that are associated with your AWS account.
- SSM Incidents – Provides `Get` and `List` permissions for SSM Incidents resources that are associated with your AWS account.
- Tag – Provides `GetResources` permission for querying tagged resources that are associated with your AWS account.

The AWS managed policy `AWSResilienceHubV2AssessmentExecutionPolicy` provides these permissions. Attach the managed policy rather than copying the policy document so

that your permissions stay current as AWS updates it. To view the full policy document, see AWSResilienceHubV2AssessmentExecutionPolicy in the AWS Managed Policy Reference Guide.

AWSResilienceHubResilienceTestingPolicy

You can attach the `AWSResilienceHubResilienceTestingPolicy` to your IAM identities. This policy grants Resilience Hub the AWS Fault Injection Service (AWS FIS) permissions needed to start and manage experiments on your behalf during resilience testing. Experiments started by Resilience Hub are tagged with `managedBy: resiliencehub`.

Permission details

This policy includes the following permissions:

- AWS Fault Injection Service (AWS FIS) – Provides permissions to author and manage the experiment templates and experiments that Resilience Hub runs on your behalf: creating and deleting experiment templates, starting and stopping experiments, monitoring experiment state, listing the customer resources targeted by an experiment, tagging resources on create, and configuring multi-account targets. These actions are scoped to resources tagged with `managedBy: resiliencehub`.
- IAM – Provides `iam:PassRole` to pass the test execution role to AWS FIS, and `iam:CreateServiceLinkedRole` to allow AWS FIS to create the service-linked role it needs to run experiments.
- AWS Application Recovery Controller (ARC) Region switch – Provides `List` and `Get` permissions to monitor Region switch plan executions that run as part of an experiment.
- Amazon CloudWatch (CloudWatch) – Provides `DescribeAlarmHistory` to evaluate customer alarms as part of the post-experiment workflow during resilience testing.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Sid": "AWSResilienceHubFISActionStatement",
      "Effect": "Allow",
      "Action": "fis:CreateExperimentTemplate",
      "Resource": "arn:aws:fis:*:*:action/*"
    },
    {
      "Sid": "AWSResilienceHubFISCreateExperimentTemplateStatement",
```

```
      "Effect": "Allow",
      "Action": "fis:CreateExperimentTemplate",
      "Resource": "arn:aws:fis:*:*:experiment-template/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedBy": "resiliencehub"
        }
      }
    },
    {
      "Sid": "AWSResilienceHubFISStartExperimentFromTemplateStatement",
      "Effect": "Allow",
      "Action": "fis:StartExperiment",
      "Resource": "arn:aws:fis:*:*:experiment-template/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/managedBy": "resiliencehub"
        }
      }
    },
    {
      "Sid": "AWSResilienceHubFISStartExperimentStatement",
      "Effect": "Allow",
      "Action": "fis:StartExperiment",
      "Resource": "arn:aws:fis:*:*:experiment/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedBy": "resiliencehub"
        }
      }
    },
    {
      "Sid": "AWSResilienceHubFISExperimentStatement",
      "Effect": "Allow",
      "Action": [
        "fis:GetExperiment",
        "fis:StopExperiment",
        "fis:ListExperimentResolvedTargets"
      ],
      "Resource": "arn:aws:fis:*:*:experiment/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/managedBy": "resiliencehub"
        }
```

```
      }
    },
    {
      "Sid": "AWSResilienceHubFISExperimentTemplateStatement",
      "Effect": "Allow",
      "Action": [
        "fis:CreateTargetAccountConfiguration",
        "fis:DeleteExperimentTemplate"
      ],
      "Resource": "arn:aws:fis:*:*:experiment-template/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/managedBy": "resiliencehub"
        }
      }
    },
    {
      "Sid": "AWSResilienceHubFISPassRoleStatement",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::*:role/*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "fis.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AWSResilienceHubFISTagResourceStatement",
      "Effect": "Allow",
      "Action": "fis:TagResource",
      "Resource": [
        "arn:aws:fis:*:*:experiment-template/*",
        "arn:aws:fis:*:*:experiment/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedBy": "resiliencehub"
        }
      }
    },
    {
      "Sid": "AWSResilienceHubRegionSwitchStatement",
      "Effect": "Allow",
```

```
      "Action": [
        "arc-region-switch:ListPlanExecutions",
        "arc-region-switch:GetPlanExecution"
      ],
      "Resource": "arn:aws:arc-region-switch::*:plan/*:*"
    },
    {
      "Sid": "AWSResilienceHubFISCreateSLRStatement",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "fis.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AWSResilienceHubCloudWatchAlarmStatement",
      "Effect": "Allow",
      "Action": "cloudwatch:DescribeAlarmHistory",
      "Resource": "*"
    }
  ]
}
```

AWSResilienceHubServiceRolePolicy

The `AWSResilienceHubServiceRolePolicy` is a service-linked role (SLR) policy. You cannot attach this policy to your IAM entities. Next generation Resilience Hub automatically creates this service-linked role when you enable AWS Organizations support for multi-account resilience management. This role trusts the `resiliencehub.amazonaws.com` service principal.

For more information about service-linked roles for AWS Resilience Hub, see Using service-linked roles for AWS Resilience Hub.

Permission details

With this policy, Next generation Resilience Hub can access read-only AWS Organizations information for multi-account resilience management.

This policy includes the following permissions:

- AWS Organizations – Provides `Describe` and `List` permissions for Organizations resources. These permissions discover organization structure, identify delegated administrators, and verify trusted access status.

The following IAM policy provides required permissions for Next generation Resilience Hub to access AWS Organizations resources for multi-account resilience management.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSResilienceHubOrganizationsReadStatement",
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeAccount",
                "organizations:DescribeOrganization",
                "organizations:DescribeOrganizationalUnit",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListAccounts",
                "organizations:ListAccountsForParent",
                "organizations:ListChildren",
                "organizations:ListDelegatedAdministrators",
                "organizations:ListDelegatedServicesForAccount",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:ListParents",
                "organizations:ListRoots",
                "organizations:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Next generation Resilience Hub updates to AWS managed policies

View details about updates to AWS managed policies for Next generation Resilience Hub since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Next generation Resilience Hub Document history page.

Change	Description	Date
AWSResilienceHubResilienceTestingPolicy – New policy	AWS added a new policy for the next generation of Resilience Hub to grant the AWS Fault Injection Service (AWS FIS) permissions needed to start and manage experiments on your behalf during resilience testing.	July 31, 2026
AWSResilienceHubServiceRolePolicy – Update to an existing policy	Next generation Resilience Hub added read-only AWS Organizations permissions to the `AWSResilienceHubServiceRolePolicy` . These permissions support multi-account resilience management, including discovering organization structure, identifying delegated administrators, and verifying trusted access status.	July 7, 2026
AWSResilienceHubV2AssessmentExecutionPolicy – New policy	Next generation Resilience Hub added a new policy to grant read-only access permissions to other AWS services for resilience discovery, assessment, and management.	June 18, 2026
Next generation Resilience Hub started tracking changes	Next generation Resilience Hub started tracking changes for its AWS managed policies.	June 18, 2026

Configuring AWS Organizations integration (optional)

If you use AWS Organizations, you can optionally configure the next generation of Resilience Hub to provide centralized governance across your organization. The following is a quick setup summary:

1. The management account enables trusted access for `resiliencehub.amazonaws.com`.
2. Service-Linked Roles (`AWSServiceRoleForResilienceHub`) are automatically created in all member accounts.
3. The management account registers a delegated administrator.
4. The delegated administrator selects a home Region for data aggregation.

Individual service owners in member accounts still create their own invoker roles for their services. The SLR provides read-only cross-account visibility to the delegated administrator.

Setting a home Region

If you use AWS Organizations, the delegated administrator selects a home Region where organization-level data is aggregated. The home Region is the AWS Region where all organization-level summary data is collected for centralized reporting and dashboards.

When selecting a home Region, choose a Region that:

- Is close to your primary operations team.
- Meets your data residency requirements.

Service summary data (identifiers, compliance scores, status) replicates from all Regions and accounts into the home Region for fast, centralized queries. Full details such as topology, findings, and dependencies remain in their source Regions and are accessed on demand.

For detailed setup instructions, see AWS Organizations integration.

Multi-account setup (without AWS Organizations)

If your service's resources span multiple AWS accounts and you are not using AWS Organizations, you need cross-account roles in addition to the invoker role.

Step 1: Create cross-account roles

In each account that contains resources for your service, create a role with:

- `AWSResilienceHubV2AssessmentExecutionPolicy` attached.
- A trust policy that allows the invoker role to assume it, using an `ExternalId` to prevent confused deputy attacks. Use a unique `ExternalId` value per service and account combination:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "resiliencehub.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  },
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::123456789012:role/AWSResilienceHubAssessmentRole"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "sts:ExternalId": "ngrh-my-service-111122223333"
      }
    }
  }]
}
```

Step 2: Grant the invoker role permission to assume cross-account roles

Add an inline policy to your invoker role that allows it to assume the cross-account roles:

```
{
  "Effect": "Allow",
  "Action": "sts:AssumeRole",
  "Resource": [
    "arn:aws:iam::111122223333:role/NGRHResourceRole",
    "arn:aws:iam::444455556666:role/NGRHResourceRole"
  ]
}
```

Step 3: Configure cross-account roles on your service

Specify the cross-account role ARNs and external IDs when creating the service:

```
aws resiliencehubv2 create-service \
  --name "my-service" \
  --regions '["us-east-1"]' \
  --permission-model '{
    "invokerRoleName": "AWSResilienceHubAssessmentRole",
    "crossAccountRoles": [
      {
        "crossAccountRoleArn": "arn:aws:iam::111122223333:role/NGRHResourceRole",
        "externalId": "ngrh-my-service-111122223333"
      },
      {
        "crossAccountRoleArn": "arn:aws:iam::444455556666:role/NGRHResourceRole",
        "externalId": "ngrh-my-service-444455556666"
      }
    ]
  }'
```

You can configure up to 5 cross-account role ARNs per service.

Customer-managed keys (optional)

Next generation Resilience Hub supports customer-managed AWS KMS keys (CMKs) for encrypting your data. To use a CMK, ensure your IAM policy includes the following AWS KMS permissions:

- `kms:DescribeKey`
- `kms:GenerateDataKey`
- `kms:Encrypt`
- `kms:Decrypt`

For scheduled or long-running assessments, also include `kms:CreateGrant`.

No changes to the invoker role are needed for CMK encryption. Next generation Resilience Hub uses your caller identity for synchronous operations and AWS KMS grants for asynchronous operations.

Getting started with Next generation Resilience Hub

This tutorial walks you through the end-to-end workflow for using the next generation of Resilience Hub, from creating your first service to reviewing failure mode findings.

Topics

- Tutorial overview and prerequisites
- Step 1: Configure a resilience policy
- Step 2: Create your first system and service
- Step 3: Run your first failure mode assessment
- Step 4: Review failure mode findings and recommendations
- Step 5: Enable dependency discovery (optional)
- Step 6: Run your first resilience test (optional)

Tutorial overview and prerequisites

This tutorial walks you through the end-to-end workflow for creating your first system and service, running a failure mode assessment, and reviewing the results. By the end, you will have a working resilience assessment for one of your applications.

Step	What you will do	Estimated time
1	Configure a resilience policy	3 minutes
2	Create your first system and service	5 minutes
3	Run your first failure mode assessment	5–15 minutes (asynchronous)
4	Review findings and recommendations	10 minutes
5	Enable dependency discovery (optional)	5 minutes

Step	What you will do	Estimated time
6	Run your first resilience test (optional)	10 minutes

Total time: approximately 40–50 minutes.

Before you begin this tutorial, ensure the following:

- You have an AWS account with deployed resources (AWS CloudFormation stack, tagged resources, or Amazon EKS cluster).
- You have an IAM service role for the next generation of Resilience Hub. You can let the console create one automatically when you create your service, or create one manually beforehand. For more information, see Required IAM permissions and roles.
- You have IAM permissions to call the next generation of Resilience Hub APIs.

Step 1: Configure a resilience policy

A resilience policy defines the resilience targets for your service, such as availability SLO targets, recovery time objectives (RTO), and recovery point objectives (RPO). In this step, you create a policy and apply it to your service.

Console:

1. Choose **Policies** > **Create policy**.
2. Enter a name (for example, `Standard Availability`).
3. Enable **Availability SLO** and set it to `99.9%`.
4. (Optional) Enable **Multi-Region DR** if your application spans Regions.
5. Choose **Create**.
6. Navigate to your service and apply the policy.

AWS CLI:

```
# Create the policy
aws resiliencehubv2 create-policy \
  --name "standard-availability" \
```

```
  --availability-slo '{"target": 99.9}'

# Apply the policy to your service
aws resiliencehubv2 update-service \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/api-
service:def456" \
  --policy-arn "arn:aws:resiliencehub:us-east-1:123456789012:policy/standard-
availability:ghi789"
```

For more information about policies, see Resilience policies.

Step 2: Create your first system and service

A system in the next generation of Resilience Hub represents a business application. A service represents a deployable component within that system. In this step, you create your first system and associate a service with it.

Create a system

Console:

1. Open the the next generation of Resilience Hub console.
2. Choose **Systems** > **Create system**.
3. Enter a name (for example, `My Application`) and an optional description.
4. Choose **Create**.

AWS CLI:

```
aws resiliencehubv2 create-system \
  --name "my-application" \
  --description "My first application in Resilience Hub"
```

Create a service

A service represents a deployable component. When you create a service, you associate it with your system and specify where the next generation of Resilience Hub should look for your AWS resources (input sources).

Console:

1. Choose **Services** > **Create service**.
2. Enter a name (for example, `api-service`).
3. Select your system.
4. Under **Permission model**, do one of the following:
   - To let the console create the required role automatically, choose **Create new role**.
   - If you already have a role, choose **Use an existing service role** and choose it from the list.
5. Under **Input sources**, add your AWS CloudFormation stack ARN, resource tags, or Terraform state file location.
6. Choose **Create**.

AWS CLI:

```
aws resiliencehubv2 create-service \
  --name "api-service" \
  --regions '["us-east-1"]' \
  --associated-systems '[{"systemArn": "arn:aws:resiliencehub:us-
east-1:123456789012:system/my-application:abc123"}]' \
  --permission-model '{"invokerRoleName": "AWSResilienceHubAssessmentRole"}'
```

Associate service with system

After creating your system and service, associate the service with the system by creating a user journey.

Console:

1. Choose **Systems** and select your system name.
2. Choose **Create user journey**.
3. Enter a name for the user journey (for example, `checkout process`).
4. Select the service you created to associate it with this user journey.
5. Choose **Create user journey**.

Step 3: Run your first failure mode assessment

With a service created and a policy applied, you can now run a failure mode assessment. During the assessment, the next generation of Resilience Hub automatically discovers your resources and builds a topology before analyzing failure modes.

Console:

1. Navigate to your service.
2. Choose **Failure mode guidance** and add any assertions about your service. For more information, see Failure mode guidance.
3. Choose **Run failure mode assessment**.
4. Wait for the assessment to complete. This typically takes 5–15 minutes.

AWS CLI:

```
aws resiliencehubv2 start-failure-mode-assessment \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/api-
service:def456"
```

To check the status of the assessment:

```
aws resiliencehubv2 list-failure-mode-assessments \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/api-
service:def456"
```

What happens during the assessment

While the assessment runs, the next generation of Resilience Hub performs the following steps in the background:

1. the next generation of Resilience Hub assumes your invoker role.
2. It reads resources from your configured input sources (AWS CloudFormation, tags, Terraform, Amazon EKS).
3. It identifies parent-child relationships (for example, Auto Scaling group to EC2 instances).
4. Resilience Hub builds a topology of your service.
5. It builds a topology showing data flow and containment.

6. A multi-agent AI system analyzes the architecture for failure modes against your resilience policy and AWS Well-Architected best practices.

Once topology generation is complete, you can view the results in the console:

- **Graph view** – Visual map of resources and connections.
- **Table view** – List of all discovered resources with metadata.
- **JSON export** – Download the full topology for external analysis.

Step 4: Review failure mode findings and recommendations

Once the assessment completes, review the results to understand the potential failure modes in your service and the recommended improvements.

Console:

1. Navigate to your service and choose the **Failure modes** tab.
2. Review findings sorted by severity.
3. Choose any finding to see detailed reasoning and recommendations.

AWS CLI:

```
aws resiliencehubv2 list-failure-mode-findings \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/api-
service:def456"
```

Understanding your findings

Each finding tells you:

- **What** the failure mode is (for example, "Single-AZ RDS database").
- **Why** it matters for your architecture.
- **How** to fix it, with recommendations that include cost and complexity considerations.
- **Which policy** requirement it relates to.

Taking action

For each finding, you can:

- **Reason** – Think about the failure mode in detail and reason about the likelihood and impact of it occurring.
- **Fix it** – Implement the recommendation if the likelihood and impact warrant it, then re-run the assessment to verify.
- **Mark as irrelevant** – Suppress the finding if it does not apply to your use case. Suppressed findings remain suppressed across future assessments.
- **Prioritize** – Use severity to decide what to address first.

Step 5: Enable dependency discovery (optional)

Dependency discovery automatically identifies the AWS services, internal endpoints, and third-party endpoints that your service calls. It uses DNS query log analysis with a 35-day look-back window and continuous polling to identify dependencies you may not know about, including unexpected cross-Region calls or critical third-party dependencies.

Dependency discovery is an optional add-on. To enable it, navigate to your service in the console and choose **Enable dependency discovery**. Once enabled, the next generation of Resilience Hub begins discovering dependencies and surfaces them on the **Dependencies** tab of your service.

For more information, see Dependency discovery.

Step 6: Run your first resilience test (optional)

Resilience testing helps validate that your service operates and recovers as expected by injecting controlled faults using AWS Fault Injection Service (AWS FIS). In this step, you choose a test template, create a test for your service, and start a test run.

Console

1. Navigate to your service and choose the Testing tab. You will see recommended resilience tests based on your service's architecture and resilience policy.
2. Select a test template (for example, **Availability Zone: recovery**). Choose **View details** to learn more about the recommended test.
3. Choose Edit test configuration, provide any required parameters, and save. This creates your test from the template - your configuration is saved and reused for future runs. You can edit your test configuration anytime.

4. Choose Start test to start a test run.
5. Monitor the run and review whether your recovery objectives were met.

AWS CLI

```
# List the available test templates
aws resiliencehubv2 list-test-templates --region us-east-1

# Create a test for your service from a template
aws resiliencehubv2 create-test \
  --test-template-arn "arn:aws:resiliencehub:us-east-1:aws:test-template/aws-az-
recovery:rtaz001" \
  --target-identifier "arn:aws:resiliencehub:us-east-1:123456789012:service/api-
service:def456" \
  --parameters '{"availabilityZone": ["us-east-1a"]}'

# Start a test run
aws resiliencehubv2 start-test-run --test-id "test-id"

# Check the status and results of the test run
aws resiliencehubv2 get-test-run --test-run-id "test-run-id"
```

For more information, see Resilience testing.

Systems, user journeys, and services

Application modeling is the foundation of the next generation of Resilience Hub. You model your applications using a hierarchy of systems, user journeys, and services that matches how your organization thinks about its business applications.

Topics

- Overview: the systems, user journeys, and services hierarchy
- Creating and managing systems
- Creating and managing user journeys
- Creating and managing services
- Adding and removing resources from a service
- Importing application context from external registries
- Viewing service topology

- Example: Modeling a multi-account application

Overview: the systems, user journeys, and services hierarchy

The application modeling hierarchy in the next generation of Resilience Hub consists of three levels:

- **Systems** – The top-level container representing a business application (for example, an e-commerce platform or a trading system).
- **User journeys** – Critical business paths within a system (for example, "Path to purchase" or "Order fulfillment").
- **Services** – The building blocks comprising AWS resources, code, and observability that support user journeys. A service belongs to exactly one system but can support multiple user journeys.

The following example shows a hierarchy for an e-commerce platform:

```
System (e.g., "E-commerce Platform")
### User Journey: "Path to purchase"
#   ### Service: "Auth Service"
#   ### Service: "Checkout Service"
#   ### Service: "Payment Service"
### User Journey: "Order fulfillment"
#   ### Service: "Order Management Service"
#   ### Service: "Shipping Service"
### Shared Service: "EKS Platform Service"
    (supports multiple user journeys)
```

You define where to find your AWS resources (AWS CloudFormation stacks, Terraform state files, resource tags, or Amazon EKS clusters), and the next generation of Resilience Hub automatically discovers and maps them into a topology showing how resources connect.

How application modeling works

To model your application in the next generation of Resilience Hub, follow these steps:

1. **Create a resilience policy** – Define your availability, disaster recovery, and data recovery targets. See the section called “Resilience policies”.

2. **Create a system** – Define the top-level business application. See the section called “Create a system”.
3. **Create user journeys** – Define the critical business paths within your system. See the section called “Creating and managing user journeys”.
4. **Create services** – Define the building blocks and specify input sources for resource discovery. See the section called “Creating and managing services”.
5. **Run an assessment** – Start a failure mode assessment to identify resilience gaps. See the section called “Failure mode assessments”.

Creating and managing systems

A system represents a business application that your organization operates. Most customers create one system per major business application.

Create a system

To create a system (console)

1. Open the Next generation Resilience Hub console at https://console.aws.amazon.com/resiliencehub/v2/home.
2. In the navigation pane, choose **Systems**.
3. Choose **Create system**.
4. Provide the following details:
   - **System name** – Enter a descriptive name for your system. The name must be 1–60 characters and can contain letters, numbers, hyphens, and underscores.
   - **Description** – (Optional) Enter a description that helps your team identify the purpose of this system.
   - **Cross-account sharing** – (Optional) Enable this setting if you want services in other AWS accounts to associate with this system.
   - **Encryption** – (Optional) Choose a customer managed AWS KMS key to encrypt system data.
   - **Tags** – (Optional) Add one or more tags to your system.
5. Choose **Create system**.

To create a system (AWS CLI)

- Run the following command:

```
aws resiliencehubv2 create-system \
  --name "system-name" \
  --description "system-description"
```

When to create multiple systems

Create **separate systems** when:

- Applications or services are independently operated by different teams.
- You want separate resilience posture tracking for each application.

Use a **single system** when:

- Multiple services work together to deliver business value.
- You want to assess resilience across related services as a group.
- Services share infrastructure (for example, a common Amazon EKS platform).

List and describe systems

To view your systems (console)

1. Open the Next generation Resilience Hub console.
2. In the navigation pane, choose **Systems**.
3. The systems list shows all systems in your account with their name, number of services, and creation date.
4. To view details for a specific system, choose the system name.

To list systems (AWS CLI)

- Run the following commands:

```
aws resiliencehubv2 list-systems

aws resiliencehubv2 get-system \
  --system-arn "arn:aws:resiliencehub:region:account-id:system/system-name:id"
```

Delete a system

A system cannot be deleted while it has service associations. When you delete a system using the console, it automatically removes the service associations for you. When using the AWS CLI, you must remove all service associations first, then delete the system.

To delete a system (console)

1. Open the Next generation Resilience Hub console.
2. In the navigation pane, choose **Systems**.
3. Select the system you want to delete.
4. Choose **Actions**, then choose **Delete**.
5. In the confirmation dialog, select the deletion acknowledgement checkbox and choose **Delete**.

To delete a system (AWS CLI)

- Run the following command:

```
aws resiliencehubv2 delete-system \
  --system-arn "arn:aws:resiliencehub:region:account-id:system/system-name:id"
```

Creating and managing user journeys

User journeys describe the critical business paths within your system. They help you organize services by business capability and apply resilience policies at the business level.

Create a user journey

To create a user journey (console)

1. Open the Next generation Resilience Hub console.
2. In the navigation pane, choose **Systems**, then choose the system you want to add a user journey to.
3. Choose the **User journeys** tab.
4. Choose **Create user journey**.
5. Provide the following details:
   - **Name** – Enter a name that describes the business capability (for example, "Path to purchase" or "Order fulfillment"). Name user journeys after what an end user does, not after technical components.
   - **Description** – (Optional) Enter a description of the user journey.
   - **Associated services** – (Optional) Select the services that support this user journey. You can add services later.
6. Choose **Create user journey**.

To create a user journey (AWS CLI)

- Run the following command:

```
aws resiliencehubv2 create-user-journey \
  --system-arn "arn:aws:resiliencehub:region:account-id:system/system-name:id" \
  --name "journey-name" \
  --description "journey-description"
```

Guidelines for defining user journeys

- Name user journeys after business capabilities, not technical components.
- A user journey should represent something an end user does (for example, "Check balance," not "Database reads").
- Start with your most critical user journeys and add more over time.

- A service can belong to multiple user journeys – shared platform services often do.

The following table shows examples of systems and their corresponding user journeys:

System	User journeys
E-commerce platform	Path to purchase, Product discovery, Order fulfillment, Returns
Trading system	Execute trade, Check portfolio, Transfer funds
Failover orchestration	Manage plans, Execute failover, Run reports

Update a user journey

To update a user journey (console)

1. Open the Next generation Resilience Hub console and navigate to your system.
2. Choose the **User journeys** tab.
3. Choose the user journey you want to update.
4. Choose **Edit**.
5. Modify the name, description, or associated services as needed.
6. Choose **Save changes**.

Delete a user journey

To delete a user journey (console)

1. Open the Next generation Resilience Hub console and navigate to your system.
2. Choose the **User journeys** tab.
3. Select the user journey you want to delete.
4. Choose **Delete**.
5. Confirm the deletion.

Creating and managing services

Services are the primary entity you interact with in the next generation of Resilience Hub. A service represents a deployable unit that comprises AWS resources, code, and observability.

Create a service

To create a service (console)

1. Open the Next generation Resilience Hub console.
2. In the navigation pane, choose **Services**.
3. Choose **Create service**.
4. Provide the following details:

   - **Service name** – Enter a descriptive name for your service (for example, `checkout-service` or `payment-service`).
   - **Description** – (Optional) Enter a description of the service.
   - **Resilience policy** – Select a resilience policy to associate with this service. The policy defines your availability SLO and RTO/RPO targets.
   - **Permission model** – Specify the IAM role that the next generation of Resilience Hub uses for resource discovery. Choose one of the following options:
     - **Create a new service role** (recommended) – The console automatically creates an IAM role with the correct trust policy and attaches the required managed policy.
     - **Use an existing service role** – Choose an existing IAM role from the list. Make sure that the role has the required trust policy and permissions. For more information, see Required IAM permissions and roles.
     - **Cross-account roles** – (Optional) If your resources are in other accounts, add the cross-account role ARNs.
   - **Regions** – Select the AWS Regions where your service operates. You can select up to 5 Regions.
   - **Resource discovery** – Provide input sources to enable discovery of resources used by your service. See the section called “Add input sources to a service” for instructions on adding input sources.

- **Dependency discovery** – (Optional) Enable the dependency discovery feature for this service. See the section called "Dependency discovery" for more information on dependency discovery.
- **Data encryption** – (Optional) Choose a customer managed AWS KMS key to encrypt service data. For more information, see the section called "Data encryption".
- **Tags** – (Optional) Add tags to your service.

5. Choose **Create service**.

To create a service (AWS CLI)

- Run the following command:

```
aws resiliencehubv2 create-service \
  --name "service-name" \
  --regions '["region"]' \
  --permission-model '{"invokerRoleName": "role-name"}' \
  --associated-systems '[{"systemArn": "system-arn"}]'
```

The following table describes the available input source types that you can add after creating the service:

Input source	Use when
AWS CloudFormation stacks	Your infrastructure is defined in AWS CloudFormation.
Terraform state files	Your infrastructure is managed by Terraform (state file in Amazon S3).
Resource tags	The service discovers resources by matching tags. A single tag-based input source with multiple tags discovers only resources that match all specified tags. Multiple tag-based

Input source	Use when
	input sources discover resources that match any of them.
Amazon Elastic Kubernetes Service clusters	Your service runs on Amazon EKS. Discovery can be scoped by namespace and, optionally, by label selector.

Add input sources to a service

To add an input source (console)

1. Open the Next generation Resilience Hub console and navigate to your service.
2. Choose the **Configuration** tab.
3. Choose **Edit** in the **Service resource discovery** section.
4. Select the input source type and provide the required configuration:

   - **AWS CloudFormation stack** – Select one or more AWS CloudFormation stacks. The selector will show stacks relevant to the Regions you have selected for this service.
   - **Terraform state file** – Enter the Amazon S3 URL of the state file.
   - **Resource tags** – Enter the tag key and values to match.
   - **Amazon EKS cluster** – Enter the cluster ARN and select the namespaces to include. Optionally, add a label selector to discover only the Kubernetes objects whose labels match. For information about the selector syntax, see EksSource in the *Next generation Resilience Hub API Reference*.

   > **Note**
   >
   > the next generation of Resilience Hub supports public, private, and CIDR-restricted Amazon EKS clusters. A cluster with `endpointPublicAccess=false` requires no VPC endpoint, AWS PrivateLink, peering, or VPN to be discovered. You do not need to change your cluster's network configuration or security baseline.

5. Choose **Add**.

To add an input source (AWS CLI)

- Run the following command:

```
aws resiliencehubv2 create-input-source \
  --service-arn "service-arn" \
  --resource-configuration '{"cfnStackArn": "stack-arn"}'
```

Service placement: same account vs. different account

Same account as system (simple): The system and service are in the same AWS account. A single invoker role covers everything. This approach is best for small teams and single-account workloads.

Different account than system (enterprise): The system is in a central account and services are in spoke accounts. This approach requires cross-account roles or AWS Organizations. It is best for large organizations with distributed ownership.

Service functions

Service functions are technical subsets of the service topology that represent specific workflows within a service.

For example, an authentication service might have two service functions:

- **SSO sign-in** – involving the identity provider, session store, and token service.
- **Registration** – involving the user database, email service, and verification flow.

Service functions help you understand which resources participate in which workflows within a single service. Service functions are identified by the next generation of Resilience Hub during assessment.

Viewing the service event log

The service event log provides a chronological history of all events that have occurred for a service. You can use the event log to track changes, troubleshoot issues, and audit activity.

To view the service event log (console)

1. Navigate to your service.
2. Choose the **Actions** button.
3. Select **View event log**.

You can filter events by **Event type** and **Time range**. The following event types are recorded:

- **Assertion created** – An assertion was added to the service.
- **Assertion deleted** – An assertion was removed from the service.
- **Assertion updated** – An existing assertion was modified.
- **Achievability updated** – The achievability status of a policy component changed.
- **Service created** – The service was created.
- **Service deleted** – The service was deleted.
- **Function created** – A service function was created.
- **Function deleted** – A service function was deleted.
- **Function resources added** – Resources were added to a service function.
- **Function resources removed** – Resources were removed from a service function.
- **Function updated** – A service function was updated.
- **Input sources updated** – The service's input sources were modified.
- **Policy associated** – A resilience policy was associated with the service.
- **Policy disassociated** – A resilience policy was removed from the service.
- **Resources associated** – New resources were discovered and associated with the service.
- **Resources disassociated** – Resources were removed from the service.
- **System associated** – The service was associated with a system.
- **System disassociated** – The service was removed from a system.
- **Workflow updated** – A workflow configuration was updated.

Each event entry shows the timestamp, event type, and a summary of what changed. Choose **Show details** to view additional information about a specific event.

Adding and removing resources from a service

Resources are discovered automatically from your configured input sources. You do not manually add individual resources. the next generation of Resilience Hub discovers resources from AWS CloudFormation stacks, Terraform state files, Amazon EKS clusters, and resource tags.

View discovered resources

To view resources (console)

1. Open the Next generation Resilience Hub console and navigate to your service.
2. Choose **Service topology**.
3. The resources graph and resources list show all discovered resources with their type, Region, and associated service function.
4. Use the filters to narrow by Region or service function.

To list resources (AWS CLI)

- Run the following command:

```
aws resiliencehubv2 list-resources \
  --service-arn "service-arn"
```

To filter by Region:

```
aws resiliencehubv2 list-resources \
  --service-arn "service-arn" \
  --aws-region "region"
```

Update input sources

To change where the next generation of Resilience Hub looks for resources, add or remove input sources on the service. See the section called “Add input sources to a service” for instructions on adding input sources.

To remove an input source (console)

1. Open the Next generation Resilience Hub console and navigate to your service.
2. Choose the **Configuration** tab.
3. Choose **Edit**.
4. Remove existing input source selections.
5. Choose **Save changes**.

To remove an input source (AWS CLI)

- Run the following command:

```
aws resiliencehubv2 delete-input-source \
  --service-arn "service-arn" \
  --input-source-id "input-source-id"
```

Importing application context from external registries

You can import application context from external registries, such as design documents, to enrich your application model in the next generation of Resilience Hub. Imported context supplements the resource topology with business metadata that is not captured in AWS resource configuration.

Import design documents

Design documents provide additional architectural context that the next generation of Resilience Hub uses during failure mode assessments. You can upload design documents stored in Amazon S3 as input sources for your service.

To import a design document (console)

1. Open the Next generation Resilience Hub console and navigate to your service.
2. Choose the **Assessments** tab.
3. Choose **Failure mode guidance**.
4. Choose **Add more** from Architecture design files.
5. Enter the Amazon S3 URL of your design document.

6. Choose **Save design file**.

To import a design document (AWS CLI)

- Run the following command:

```
aws resiliencehubv2 create-input-source \
  --service-arn "service-arn" \
  --resource-configuration '{"designFileS3Url": "s3://bucket/path/to/design-doc.pdf"}'
```

Supported design document formats include PDF, Markdown, and plain text files. The AI assessment engine uses these documents to understand your intended architecture and identify gaps between design intent and actual implementation.

Viewing service topology

The Next generation Resilience Hub console provides a visual canvas view showing your service hierarchy, including resource topology and service functions.

To view service topology

1. Open the Next generation Resilience Hub console and navigate to your service.
2. Choose **Service topology**.
3. The topology view shows how your discovered resources connect to each other. Resources are grouped by service function and show edges representing dependencies.
4. Choose a resource node to view its details, including resource type, ARN, Region, and account.

Example: Modeling a multi-account application

This example shows how to model a large e-commerce platform with 30 services across 10 accounts in Next generation Resilience Hub.

To model a multi-account application

1. **Create the system in a central account.**

Start by creating a single system that represents the entire e-commerce platform. Create the system in your central governance account and enable cross-account sharing:

```
aws resiliencehubv2 create-system \
  --name "acme-ecommerce" \
  --description "Acme Corp e-commerce platform" \
  --sharing-enabled
```

2. **Create services in their respective accounts.**

   Each team creates their service in their own account, referencing the central system ARN:

```
# In account A (auth team)
aws resiliencehubv2 create-service \
  --name "auth-service" \
  --regions '["us-east-1", "us-west-2"]' \
  --permission-model '{"invokerRoleName": "AWSResilienceHubAssessmentRole"}' \
  --associated-systems '[{"systemArn": "arn:aws:resiliencehub:us-
east-1:111111111111:system/acme-ecommerce:abc123"}]'

# In account B (checkout team)
aws resiliencehubv2 create-service \
  --name "checkout-service" \
  --regions '["us-east-1", "us-west-2"]' \
  --permission-model '{"invokerRoleName": "AWSResilienceHubAssessmentRole"}' \
  --associated-systems '[{"systemArn": "arn:aws:resiliencehub:us-
east-1:111111111111:system/acme-ecommerce:abc123"}]'
```

3. **Add input sources to each service.**

   Each team adds their resource discovery configuration:

```
# Auth team adds their CloudFormation stack
aws resiliencehubv2 create-input-source \
  --service-arn "arn:aws:resiliencehub:us-east-1:222222222222:service/auth-
service:def456" \
```

```
  --resource-configuration '{"cfnStackArn": "arn:aws:cloudformation:us-
east-1:222222222222:stack/auth-prod/..."}'

# Checkout team adds their EKS cluster
aws resiliencehubv2 create-input-source \
  --service-arn "arn:aws:resiliencehub:us-east-1:333333333333:service/checkout-
service:ghi789" \
  --resource-configuration '{"eks": {"clusterArn": "arn:aws:eks:us-
east-1:333333333333:cluster/checkout-cluster", "namespaces": ["checkout"]}}'
```

4. **Define user journeys.**

   Create user journeys in the central account that group services by business capability:

```
aws resiliencehubv2 create-user-journey \
  --system-arn "arn:aws:resiliencehub:us-east-1:111111111111:system/acme-
ecommerce:abc123" \
  --name "Path to purchase" \
  --description "Customer browses, adds to cart, and completes checkout"
```

5. **Apply a resilience policy.**

   Create a policy and associate it with services:

```
# Create a policy
aws resiliencehubv2 create-policy \
  --name "mission-critical" \
  --availability-slo '{"target": 99.99}' \
  --multi-az '{"rtoInMinutes": 5, "rpoInMinutes": 1, "disasterRecoveryApproach":
 "ACTIVE_ACTIVE"}' \
  --multi-region '{"rtoInMinutes": 30, "rpoInMinutes": 5,
 "disasterRecoveryApproach": "WARM_STANDBY"}'

# Associate with a service
aws resiliencehubv2 update-service \
  --service-arn "arn:aws:resiliencehub:us-east-1:333333333333:service/checkout-
service:ghi789" \
  --policy-arn "arn:aws:resiliencehub:us-east-1:111111111111:policy/mission-
critical:xyz"
```

6. **Run assessments.**

   Start a failure mode assessment on each service to identify resilience gaps:

```
aws resiliencehubv2 start-failure-mode-assessment \
  --service-arn "arn:aws:resiliencehub:us-east-1:333333333333:service/checkout-
service:ghi789"
```

Services can be added incrementally – start with the most critical services and onboard more over time. The system-level view in the console shows all services across accounts in a single canvas.

Resilience policies

Match your organizational resilience requirements with resilience policies. A resilience policy defines your resilience expectations through modular, composable requirements. Rather than choosing a single rigid policy type, you construct policies by selecting the requirements that matter to your application.

Topics

- Understanding resilience policies
- Policy components
- Creating a resilience policy
- Applying policies at user journey and service levels
- Policy inheritance and multi-level evaluation
- Managing and updating policies
- Sharing a policy across your organization

Understanding resilience policies

Resilience policies define the resilience requirements for your application. Each policy specifies targets that Next generation Resilience Hub evaluates during failure mode assessments to determine whether your application meets your resilience goals.

Different stakeholders care about different aspects of resilience:

- **Central SRE teams** focus on disaster recovery (RTO/RPO) and availability targets.
- **Service teams** focus on operational performance (latency, error rates, throughput).
- **Compliance teams** focus on data protection (RPO for data durability).

Modular policies let each stakeholder define their requirements independently, then apply them at the appropriate level of the application hierarchy.

Policy components

Policies include the following components:

Availability service level objective (SLO)

Define the target SLO for the services associated with this policy.

Multi-Region disaster recovery

Define a multi-Region approach, target recovery time objective (RTO), and recovery point objective (RPO) for services associated with this policy.

Multi-AZ disaster recovery

Define a multi-AZ approach, target recovery time objective (RTO), and recovery point objective (RPO) for services associated with this policy.

Data recovery objective

Define the amount of time between backups for data/storage for services associated with this policy.

Creating a resilience policy

You can create resilience policies from the the next generation of Resilience Hub console or AWS CLI.

Console:

1. Navigate to **Resilience Hub** > **Policies**.
2. Choose **Create policy**.
3. Enter a policy name and description.
4. Select the components you need:
   - Toggle **Multi-Region DR** and set RTO/RPO.
   - Toggle **Availability SLO** and set target percentage.
   - Toggle **Data recovery objective** and set time between backups.
5. Choose **Create**.

AWS CLI:

```
aws resiliencehubv2 create-policy \
  --name "Critical Service Policy" \
  --multi-region '{"rtoInMinutes": 15, "rpoInMinutes": 5, "disasterRecoveryApproach":
 "WARM_STANDBY"}' \
  --availability-slo '{"target": 99.99}'
```

Applying policies at user journey and service levels

Policies can be applied at two levels of the application hierarchy:

Level	How to apply	Who typically owns it
User journey	Applied on the user journey within the system	Central SRE team
Service	Applied directly on the service entity	Service team

To apply a policy to a user journey, use the following command. All services within that user journey are evaluated against this policy:

```
aws resiliencehubv2 update-user-journey \
  --system-arn "arn:aws:resiliencehub:region:account-id:system/system-name:id" \
```

```
  --user-journey-id "user-journey-id" \
  --policy-arn "arn:aws:resiliencehub:us-east-1:123456789012:policy/critical-dr:abc123"
```

To apply a policy directly to a service, use the following command:

```
aws resiliencehubv2 update-service \
  --service-arn "arn:aws:resiliencehub:..." \
  --policy-arn "arn:aws:resiliencehub:us-east-1:123456789012:policy/high-
performance:def456"
```

Policy inheritance and multi-level evaluation

A service is evaluated against **all policies that apply to it**:

- Its directly assigned policy (if any).
- Policies inherited from user journeys it belongs to.

This enables separation of concerns:

- Central SRE teams set DR policies at the user journey level for governance.
- Service teams set performance policies at the service level for operational requirements.
- Both policies apply simultaneously to services in the user journey.

For example:

```
User Journey: "Path to Trade"
  Policy: "Trading Platform DR Policy" (Multi-Region, RTO 15 min, RPO 5 min)
  ### Service: "Order Execution"
        Policy: "Order Execution Performance Policy" (Availability 99.99%, Error <
 0.1%, p99 < 50ms)
```

The "Order Execution" service is evaluated against **both** policies. It must meet the DR requirements from the user journey policy and the performance requirements from its service-specific policy.

Conflict resolution

When multiple policies contain conflicting requirements for the same component, Next generation Resilience Hub automatically applies the **stricter value**. The following example shows how conflicting RTO values are resolved:

Source	Component	Value
User journey policy	RTO	30 minutes
Service policy	RTO	15 minutes
Applied value	**RTO**	**15 minutes (stricter)**

Managing and updating policies

You can list, update, and delete resilience policies from the the next generation of Resilience Hub console or AWS CLI.

To list all policies in your account:

```
aws resiliencehubv2 list-policies
```

To update a policy:

```
aws resiliencehubv2 update-policy \
  --policy-arn "arn:aws:resiliencehub:..." \
  --availability-slo '{"target": 99.99}'
```

Changes to a policy take effect on the next failure mode assessment for all services using that policy.

You cannot delete a policy that is currently applied to services or user journeys. Remove all associations first.

Sharing a policy across your organization

If you use AWS Organizations, you can share a resilience policy with the member accounts in your organization. Service owners in those accounts can then apply the shared policy to their own services. Your organization measures resilience against the same requirements, without each account maintaining a separate copy of the policy.

Only the delegated administrator or the management account can enable cross-account sharing. For more information about designating a delegated administrator, see AWS Organizations integration.

Sharing a policy

Enable cross-account sharing when you create the policy:

```
aws resiliencehubv2 create-policy \
  --name "Tier 1 DR Policy" \
  --multi-region '{"rtoInMinutes": 15, "rpoInMinutes": 5, "disasterRecoveryApproach":
 "WARM_STANDBY"}' \
  --sharing-enabled
```

You can also enable it on an existing policy:

```
aws resiliencehubv2 update-policy \
  --policy-arn "arn:aws:resiliencehub:us-east-1:111122223333:policy/tier-1-dr:abc123" \
  --sharing-enabled
```

Member accounts see shared policies alongside the policies they own. Each policy reports whether the account owns it or received it through organization sharing, so service owners can tell the two apart.

Stopping sharing

To stop sharing a policy, disable cross-account sharing:

```
aws resiliencehubv2 update-policy \
```

```
  --policy-arn "arn:aws:resiliencehub:us-east-1:111122223333:policy/tier-1-dr:abc123" \
  --no-sharing-enabled
```

Next generation Resilience Hub immediately stops evaluating services in member accounts against the policy. It then clears the policy from the member services that were using it, so those services no longer show an attachment that has no effect.

Service owners can apply a different policy at any time.

The policy's timeline records changes to sharing, and the account that owns the policy can review it. For more information, see Tracking changes to a shared policy.

Assessment

Next generation Resilience Hub provides two complementary assessment capabilities: automated dependency assessment to discover and classify your service dependencies, and failure mode assessments to identify potential resilience weaknesses using GenAI-powered analysis.

Topics

- Dependency discovery
- Failure mode assessments

Dependency discovery

Dependency discovery in the next generation of Resilience Hub automatically identifies all AWS services, internal endpoints, and third-party endpoints that your services depend on and assesses their location and frequency. It uses DNS query log analysis to surface dependencies you may not know about – including unexpected cross-region calls, critical third-party dependencies, and infrequently accessed services that could cause failures during incidents.

Topics

- How dependency discovery works
- Prerequisites for dependency discovery
- Enabling dependency discovery for a service
- Monitoring discovery status
- Viewing discovered dependencies

- Dependency insights
- Classifying dependencies as hard or soft
- Coverage and known limitations
- Troubleshooting dependency discovery
- Pricing

How dependency discovery works

Dependency discovery analyzes Route 53 DNS resolver query logs to identify every domain name that your service's compute resources resolve. This reveals the full set of external dependencies without requiring any agents, code changes, or instrumentation.

Feature	Detail
Lookback window	35 days of historical DNS query data
Continuous monitoring	Ongoing discovery summarized by hour
Supported dependency types	AWS services, internal endpoints, third-party endpoints
Attribution	Dependencies are attributed to specific compute resources within your service
Setup	No agents or code changes required – enable in minutes

Supported dependency types

Dependency discovery identifies the following types of dependencies:

- **AWS service endpoints** – Amazon S3, Amazon DynamoDB, SQS, and other AWS services your application calls.
- **Internal endpoints** – Services within your VPC or organization.
- **Third-party endpoints** – External services such as LaunchDarkly, Stripe, or Datadog.
- **Cross-region calls** – Dependencies that resolve to endpoints in other AWS Regions.

Each discovered dependency is categorized by type:

Type	Description	Example
AWS service	An AWS service endpoint	Amazon S3, Amazon DynamoDB, SQS
Third-party	A service outside AWS and your organization	LaunchDarkly, Stripe, Datadog
Internal	A service within your organization	Internal microservices, shared APIs

How dependency discovery differs from observability tools

Dependency discovery focuses on **resilience validation** rather than operational observability:

Capability	Observability tools (X-Ray, CloudWatch)	Next generation Resilience Hub dependency discovery
Setup	Requires SDK instrumentation or agents	No agents or code changes – agentless
Focus	Performance monitoring, latency debugging	Resilience validation, failure testing
Discovery	Reactive – discovers failures during incidents	Proactive – discovers dependenc ies before incidents
Classification	No criticality classification	Hard/soft classification
Time to value	Days or weeks of instrumen tation	Minutes to enable

Prerequisites for dependency discovery

Before you enable dependency discovery, ensure the following requirements are met:

- Your service must have compute resources (Amazon Elastic Compute Cloud instances, Amazon Elastic Container Service tasks, Amazon Elastic Kubernetes Service pods, or VPC bound Lambda functions) that make DNS queries through Route 53 resolvers.

- The compute resources must reside in a VPC configured to use Route 53 DNS resolvers.
- Next generation Resilience Hub discovers either Amazon Elastic Compute Cloud instances (preferred) or Amazon VPC (fallback). This can be verified by viewing the Service topology in the console.
- No additional IAM permissions are required beyond the standard invoker role – dependency discovery uses Next generation Resilience Hub's own service credentials to access DNS query data.

Enabling dependency discovery for a service

You can enable dependency discovery from the console or using the AWS CLI.

To enable dependency discovery for a single service (console)

Navigate to your service in the console and choose the **Assessment** tab. Then choose **Enable dependency discovery**.

To enable dependency discovery for a single service (CLI)

```
aws resiliencehubv2 update-service \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/checkout:abc123"
 \
  --dependency-discovery "ENABLED"
```

After you enable dependency discovery, Next generation Resilience Hub performs the following steps:

1. Queries 35 days of historical DNS data for your service's compute resources.
2. Identifies, deduplicates, and attributes dependencies to your service.
3. Continues monitoring hourly to detect new dependencies and update existing ones.
4. Displays results for your service in the console under the "Assessments" tab.

Monitoring discovery status

After you enable dependency discovery, the `DependencyDiscoveryConfig` object returned by the `GetService` and `ListServices` API operations includes fields that indicate whether eligible resources have been found and whether dependency analysis is still in progress.

The following table describes the fields that indicate the current state of dependency discovery.

Field	Type	Description
`status`	String	The current state of dependency discovery: `INITIALIZING` , `ENABLED`, or `DISABLED`.
`eligibleResourceCount`	Integer	The number of compute resources eligible for dependency discovery. This value is `null` until the resource discovery process completes its first run, and is updated on each subsequent run.
`message`	String	A message describing the current state of discovery. This value is `null` when discovery is complete and dependencies are ready to view.

The `message` field provides context about the current state of discovery based on the status and the number of eligible resources. The following table shows the possible messages.

Status	Eligible resource count	Message
`INITIALIZING`	Not yet available	"Discovering resources"
`INITIALIZING`	0	"No eligible resources discovered"
`INITIALIZING`	1 or more	"Discovering dependencies"
`ENABLED`	0	"No eligible resources discovered"
`ENABLED`	1 or more	None. Dependencies are ready to view.
`DISABLED`	Any	"Enable dependency discovery to display dependencies."

The following example shows how to check discovery status using the AWS CLI.

```
aws resiliencehubv2 get-service \
```

```
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/checkout:abc123"
```

The response includes the `dependencyDiscovery` object:

```
{
  "service": {
    ...
    "dependencyDiscovery": {
      "status": "INITIALIZING",
      "updatedAt": "2026-06-30T10:00:00Z",
      "eligibleResourceCount": 5,
      "message": "Discovering dependencies"
    }
  }
}
```

When the `message` field is `null` and `status` is `ENABLED`, dependency discovery is complete and results are available through the `ListDependencies` operation.

> **Note**
>
> If `eligibleResourceCount` is 0 after discovery completes, make sure that your service's compute resources meet the prerequisites described in Prerequisites for dependency discovery.

Viewing discovered dependencies

After dependency discovery is enabled, you can view a table of discovered dependencies with the following columns:

Column	Description
Dependency name	Identified name or domain name (for example, "LaunchDarkly" or "api.stripe.com")
Type	AWS service, third-party, or internal
Location	Where the dependency is hosted (for example, AWS us-east-1, Azure us-central, or RFC1918)

Column	Description
Criticality	Hard or soft – user-assigned classification
Query frequency	Visualization of query volume over time
First seen	When the dependency was first discovered
Last seen	Most recent query to this dependency

Dependency timeline

The dependency timeline shows when each dependency was first discovered and its activity over time. Select any dependency to view the underlying domain names and destination IP addresses. Usage frequency is available over the past 35 days, shown at hourly detail for single-day windows or daily detail for weekly windows.

Compute resource attribution

Next generation Resilience Hub attributes discovered dependencies to the specific compute resources – such as Amazon EC2 instances, Lambda functions, or containers – that make DNS queries to those dependencies. You can filter the dependency list by service (when viewing at the system level), criticality (hard, soft, or unclassified), type (AWS service, third-party, or internal), or location (AWS, third-party, or internal).

Dependency insights

After Dependency discovery is enabled, a single service can have hundreds of discovered dependencies, and reviewing that list manually to find resilience risks is time consuming. With Dependency insights, a generative AI-powered capability, you can highlight meaningful patterns across your discovered dependencies and address resilience risks without manually reviewing the full dependency list. Use Dependency insights when a service has accumulated more dependencies than you can practically review by hand.

Dependency insights produces a short summary that surfaces the following:

- **Cross-Region dependencies** – Dependencies that resolve in a different AWS Region than the caller, which add latency and increase blast radius if the remote Region degrades.

- **New dependencies** – Dependencies that first appeared in the last 7 days, so that you can confirm they are intentional and validate them for resilience before traffic scales.
- **Third-party dependencies** – Non-AWS dependencies, such as Datadog, PagerDuty, or Stripe, which carry availability risk outside your control.
- **Uneven usage patterns** – Dependencies with strong day-of-week or time-of-day variance, which may indicate undocumented scheduled jobs or batch tasks.

To keep the summary focused, Dependency insights filters out common AWS service noise – such as CloudWatch, NTP, and SSM – that does not typically indicate a resilience risk.

> **Note**
>
> Dependency insights is available only for services that have Dependency discovery enabled and have completed an initial discovery run. For more information, see Enabling dependency discovery for a service.

Generating dependency insights

Next generation Resilience Hub analyzes up to 35 days of discovered dependency data to generate insights. You can generate insights from the console or by using the AWS CLI.

To generate insights (console)

Navigate to your service in the console and choose the **Dependencies** section. Then choose **Generate insights**. When insights already exist, choose **Generate insights** again to regenerate them.

To generate insights (CLI)

```
aws resiliencehubv2 start-dependency-insights \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/checkout:abc123"
```

The response returns the current generation `status`:

```
{
  "status": "IN_PROGRESS"
}
```

> **Note**
>
> You can regenerate insights for a service once every 24 hours. Because new dependency data is summarized hourly, regenerating within the same day is unlikely to change the results. In the console, the **Generate insights** button is disabled until the rolling 24-hour window has passed.

Viewing dependency insights

After insights are generated, they are stored with the service and remain available until you regenerate them. The console displays the summary in the **Dependencies** section, along with the time the summary was last generated so that you can gauge whether it is stale.

To retrieve the stored insights using the AWS CLI, use the following command.

```
aws resiliencehubv2 get-dependency-insights \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/checkout:abc123"
```

The response includes an overview, a list of individual insights, the generation status, and the time the summary was last created.

```
{
  "status": "COMPLETED",
  "lastCreatedAt": "2026-09-15T10:00:00Z",
  "overview": "...",
  "insights": [
    {
      "category": "CROSS_REGION",
      "description": "..."
    },
    {
      "category": "NEW_DEPENDENCY",
      "description": "..."
    }
  ]
}
```

The following table describes the fields in the response.

Field	Type	Description
`status`	String	The state of insights generation: `IN_PROGRESS` , `COMPLETED` , or `FAILED`.
`overview`	String	A short narrative summary of the most significant patterns across your dependencies. Empty while status is `IN_PROGRESS` or `FAILED`.
`insights`	List	A list of individual insights, each with a `category` and a `description` . Empty while status is `IN_PROGRESS` or `FAILED`.
`lastCreatedAt`	Timestamp	When the summary was last successfully generated .

Each insight is assigned one of the following categories.

Category	Description
`CROSS_REGION`	A dependency that resolves in a different AWS Region than the caller.
`NEW_DEPENDENCY`	A dependency that first appeared in the last 7 days.
`THIRD_PARTY`	A non-AWS dependency.
`UNEVEN_USAGE`	A dependency with an irregular usage pattern.

Considerations

- Dependency insights highlights patterns and potential risks. It does not prescribe specific fixes. Validate each insight against your application before acting on it.
- For services with a large number of dependencies, insights prioritize the most significant patterns and risks rather than listing every dependency. A dependency might not appear in the summary even though it still appears in the full dependency list.

- Insights are generated from discovered dependency data. Dependencies that are not captured by Dependency discovery – for example, direct IP connections – are not reflected. For more information, see Coverage and known limitations.
- You can regenerate insights once every 24 hours per service.

Classifying dependencies as hard or soft

After reviewing discovered dependencies, classify each one to indicate its failure impact:

```
aws resiliencehubv2 update-dependency \
  --service-arn "arn:aws:resiliencehub:..." \
  --dependency-id "dependency-id" \
  --criticality "HARD" \
  --comment "Payment processing fails completely without Stripe"
```

Use the following guidelines to classify dependencies:

Classify as	When	Example
Hard	Your service fails completely if this dependency is unavailable	Primary database, payment gateway, authentication service
Soft	Your service degrades but continues operating without this dependency	Analytics API, feature flags, recommendation engine

Coverage and known limitations

Dependency discovery covers all DNS queries made through Route 53 resolvers in your VPC, including both IPv4 and IPv6 queries, and queries from Amazon EC2, Amazon ECS, Amazon EKS, and VPC-connected Lambda. The following known limitations apply:

Limitation	Impact	Workaround
Non-VPC Lambda	AWS Lambda functions without VPC connectivity are not covered	Connect AWS Lambda functions to VPC and mark the VPC as an input source for the service

Limitation	Impact	Workaround
Direct IP connections	Connections made by IP address (not DNS) are not discovered	No workaround
Infrequent dependencies	Dependencies called less than once per hour may be missed in initial discovery	35-day lookback catches most; very rare calls may not appear
Kubernetes shared tenancy	Multi-tenant Amazon EKS clusters may attribute dependencies to the wrong service	Verify compute resource attribution is correctly mapping resources to services

Troubleshooting dependency discovery

If dependency discovery is not returning expected results, review the following common issues:

- **Discovery not returning expected dependencies for EC2 instances** – Verify the dependency uses DNS resolution (not a direct IP), confirm that the compute resources are in a VPC with a Route 53 resolver, and check that the dependency was called within the 35-day lookback window.
- **Discovery not returning expected dependencies for ECS Fargate** – Verify that Service topology includes the VPC for the ECS Fargate Service.
- **Discovery not returning expected dependencies for Lambda functions** – Verify that the Lambda functions are VPC-connected, and that Service topology includes the VPC.
- **Unexpected cross-region dependencies** – Cross-region dependencies are flagged automatically in the console. Common causes include hardcoded regional endpoints in application code, AWS SDK clients configured for a specific region, and third-party services routing to non-local endpoints.
- **Dependencies appearing for the wrong service** – This typically occurs with shared VPCs or Amazon EKS clusters. Verify that compute resource attribution is correctly mapping resources to services.

For additional troubleshooting guidance, see Troubleshooting Next generation Resilience Hub.

Pricing

Dependency discovery is an optional add-on:

Component	Price
Dependency discovery	$10 per service per month

This provides continuous, automated identification of all dependencies with a 35-day lookback.

Failure mode assessments

Failure mode assessments in the next generation of Resilience Hub use GenAI-powered analysis to identify potential failure modes in your services and provide actionable recommendations. Assessments evaluate your architecture against your defined resilience policies, AWS Well-Architected best practices, and the AWS Resilience Analysis Framework.

Topics

- How failure mode assessments work
- Failure mode guidance
- Running a failure mode assessment at the service level
- Viewing failure mode findings and recommendations
- Assessment history and trends
- Pricing

How failure mode assessments work

When you run a failure mode assessment, Next generation Resilience Hub performs the following steps:

1. **Reads current resource state** – Refreshes your service's resource configuration from your AWS account.
2. **Analyzes the topology** – A multi-agent AI system examines how your resources connect and interact.
3. **Evaluates against policies using the resilience analysis framework** – Compares your architecture against your resilience policies. It first performs an assessment to determine if

policy components are achievable or not. This assessment provides directional guidance to help you understand whether the current policy's requirements are achievable with your service's current architecture. Do not use it as a guarantee or as evidence. If you need evidence, run a resilience test. For more information, see Resilience testing.

4. **Applies AWS Well-Architected best practices** – Checks for common resilience anti-patterns.
5. **Generates findings** – Identifies failure modes with severity, reasoning, and recommendations, and maps results to your resilience policies.

The assessment engine uses specialized AI agents that apply AWS Well-Architected Framework reliability best practices and the AWS Resilience Analysis Framework to your specific architecture. Agents analyze different aspects of resilience:

- **Availability** – Single points of failure, AZ distribution, and redundancy.
- **Disaster recovery** – Cross-region capabilities, replication, and failover readiness.
- **Dependency resilience** – Impact of dependency failures on your service.
- **Observability** – Monitoring gaps that could delay failure detection.

The failure mode assessment does not consume all available resources. Instead it evaluates a subset of resources known as assessed resources.

Assessed resource: A top-level infrastructure or service component that is directly evaluated during a resilience assessment. A resource is assessed if its configuration has a meaningful impact on availability, recoverability, or fault tolerance of the service. Resources outside of this scope will not have any impact on assessment and will not be surfaced in list-resources.

Relationship to resilience policies

Failure mode assessments evaluate your application against the targets defined in your resilience policies. Each finding maps to specific policy requirements, showing you exactly which resilience targets are at risk. For example:

```
Finding: "RDS database is not configured for Multi-AZ"
  -> Policy requirement: Multi-Region DR (RTO 15 min)
  -> Impact: Regional failover would require manual database restoration,
     exceeding the 15-minute RTO target

Finding: "No health check configured on ALB target group"
```

```
   -> Policy requirement: Availability SLO (99.99%)
   -> Impact: Unhealthy instances continue receiving traffic,
      degrading availability below target
```

Failure mode guidance

A failure mode assessment can observe how your resources are configured and connected, but it cannot observe how your service is used, operated, or intended to behave. With failure mode guidance, you can supply that missing context. The assessment then produces failure mode findings that are relevant to your service. Next generation Resilience Hub supports two forms of failure mode guidance:

- **Assertions** – Short statements of fact about your service that the assessment treats as true. For more information, see Assertions.
- **Service design files** – Architecture documents stored in Amazon S3 that describe your intended design. For more information, see Service design files.

Topics

- Assertions
- Service design files

Assertions

An assertion is a short statement of fact about your service, typically one or two sentences and up to 1,000 characters. It describes something a failure mode assessment cannot determine from resource configuration alone. Examples include your expected traffic volume, whether the environment is production, the disaster recovery strategy you intend to use, or a component that runs on your infrastructure but is not an AWS resource.

Assertions belong to a service. Every failure mode assessment you run for that service uses the current set of assertions. Each assertion has one of two sources:

- **AI-generated** (`AI_GENERATED`) – Generated by the assessment engine during the first failure mode assessment of a service. These assertions make explicit what the engine inferred about your service so that you can review and correct them. The engine does not generate assertions that duplicate or contradict assertions you have already added.

- **User-created** (USER) – Added by you, or created when you edit an AI-generated assertion. When you edit an AI-generated assertion, its source changes to USER and the next generation of Resilience Hub never modifies it again.

Each assertion also records when it was created and last updated. You can view, add, edit, and delete assertions in the console or with the AWS CLI at any time, before or after running an assessment.

How assessments use assertions

A failure mode assessment treats assertions as authoritative facts about your environment. Assertions take precedence over what the assessment can or cannot observe in your resource configuration. During an assessment, the next generation of Resilience Hub uses assertions to:

- **Calibrate the analysis** – For example, an assertion about peak traffic lets the assessment evaluate whether your capacity, quotas, and scaling configuration are adequate instead of speculating about load.
- **Complete the picture** – An assertion can describe a component that resource discovery cannot see, such as a self-managed ingress controller, a service mesh, or a third-party dependency, so that the assessment reasons about it as part of your architecture.
- **Suppress irrelevant failure mode findings** – If an assertion states that a capability exists or that a concern does not apply to your service, the assessment does not produce failure mode findings that contradict it. For example, asserting that a third-party tool manages database backups suppresses failure mode findings about missing backup configuration.
- **Identify service functions** – Assertions about how customers use your service help the assessment identify and describe the service functions it evaluates.

> **Important**
>
> The assessment treats assertions as facts. An inaccurate assertion can therefore hide a real risk or produce failure mode findings that do not apply to your service. Review AI-generated assertions after your first assessment, correct anything that is wrong, and keep assertions current as your service changes.

Assertions describe facts about your environment. They do not change your resilience targets. To change the availability, recovery time, or recovery point objectives that an assessment evaluates against, update your resilience policy instead. For more information, see Resilience policies.

Assertion lifecycle

The recommended workflow is to run an assessment, review the assertions it generated, refine them, and run another assessment. Assertions flow through this loop as follows:

1. **Before the first assessment (optional)** – Add any assertions you already know are important, such as your expected traffic or whether the environment is production. The first assessment uses them immediately.
2. **First assessment** – The assessment engine generates AI-generated assertions that capture what it inferred about your service. It avoids topics you have already covered with your own assertions.
3. **Review** – After the assessment completes, review the assertions. Edit any that are wrong, delete any that do not apply, and add any context that is missing.
4. **Subsequent assessments** – Each later assessment uses the current set of assertions. As long as the service has at least one assertion whose source is still `AI_GENERATED`, the assessment engine does not generate new assertions. User-created assertions are never modified or removed by the next generation of Resilience Hub.

> **Note**
>
> Editing an AI-generated assertion converts it to a user-created assertion. If you edit or delete every AI-generated assertion so that none remain, the next failure mode assessment generates a new set of AI-generated assertions. The new set avoids topics already covered by your user-created assertions. Because the new set counts toward the 20-assertion limit, delete assertions you no longer need before running that assessment. You can also use this behavior deliberately: delete all AI-generated assertions to have the engine re-evaluate your service from scratch.

Assertions are deleted when you delete the service. Assertion changes are recorded in the service event log. For more information, see Viewing the service event log.

Writing effective assertions

The most useful assertions tell the assessment something it cannot learn from your resources. Focus on the following areas:

- **Traffic and scale** – Typical and peak request rates, how spiky traffic is, and whether spikes are predictable.
- **Environment purpose** – Whether the service is production, staging, or development, and how strictly it should be assessed.
- **Components not visible to resource discovery** – Software you run on Amazon EC2 instances or Amazon EKS nodes, third-party services, on-premises systems, or dependencies in other accounts that are not part of the service's input sources.
- **Tenancy and isolation** – Whether the service is multi-tenant, how tenants are isolated, and whether Availability Zone isolation matters for your use case.
- **Disaster recovery intent** – The Multi-AZ or multi-Region strategy you intend to operate (for example, active-active or warm standby) and how data is replicated.
- **Operational practices** – Deployment strategy, backup tooling, runbooks, on-call coverage, or other practices that mitigate risk but leave no trace in resource configuration.
- **Business constraints** – Requirements such as "data loss is unacceptable" or planned changes such as an upcoming migration.

Avoid the following when writing assertions:

- **Restating resource configuration** – The assessment already knows that your database is Multi-AZ or that your Auto Scaling group has a minimum size of 2. Restating configuration adds noise without adding context.
- **Writing vague or hedged statements** – "Traffic might be high sometimes" gives the assessment nothing to calibrate against. State the fact directly.
- **Combining multiple facts** – Keep each assertion to a single topic so that you can edit or delete it independently.
- **Asserting capabilities you do not have** – If you assert a capability, the assessment suppresses failure mode findings that contradict it, so only assert mitigations that actually exist.
- **Including sensitive data** – Do not include credentials, secrets, or personal data in assertion text.

The following table shows examples of effective and less effective assertions.

Effective	Less effective	Why
"Steady-state traffic is about 500 requests per second, with predictable spikes to 5,000 requests per second during weekday business hours."	"Traffic can be high."	Specific numbers let the assessment evaluate capacity and scaling configuration. A vague statement gives it nothing to calibrate against.
"This is a production service. Any downtime affects paying customers."	"The service is important."	Stating the environment and the impact of downtime tells the assessment how strictly to evaluate the service.
"The Network Load Balancer routes all traffic to an NGINX ingress controller that runs as pods on the Amazon EKS managed node group."	"The load balancer sends traffic to the Amazon EKS cluster."	The ingress controller is software the assessment cannot see. The load balancer to cluster connection is already visible in the topology.
"Database backups are managed by a third-party backup tool with a 15-minute recovery point objective."	"Backups are handled."	Naming the mechanism and its recovery point objective lets the assessment evaluate it against your policy. "Handled" is too vague to act on.
"The service operates as active-passive across two Regions with a warm standby in us-west-2. Data is replicated asynchronously."	"The RDS instance has Multi-AZ enabled."	Disaster recovery intent is not visible in resource configuration. The Multi-AZ setting already is.

Managing assertions (console)

To manage assertions (console)

1. Open the Next generation Resilience Hub console and navigate to your service.
2. Choose the **Assessments** tab.

3. Choose **Failure mode guidance**. The assertions list shows each assertion's text, whether it was generated by the assessment engine or added by you, and when it was created and last updated.
4. Do one of the following:

   - To add an assertion, choose **Add assertion**, enter the assertion text, and save.
   - To edit an assertion, select it and edit the text inline. Editing an AI-generated assertion changes its source to user-created.
   - To remove an assertion, select it and choose **Delete**.

Changes take effect on the next failure mode assessment you run.

Managing assertions (AWS CLI)

The following examples use a service ARN in `us-east-1`. Replace the service ARN and assertion ID with your own values.

To create an assertion

```
aws resiliencehubv2 create-assertion \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/checkout:abc123"
 \
  --text "Steady-state traffic is about 500 requests per second, with predictable
 spikes to 5,000 requests per second during weekday business hours."
```

The response returns the new assertion, including its generated `assertionId`:

```
{
    "assertion": {
        "serviceArn": "arn:aws:resiliencehub:us-east-1:123456789012:service/
checkout:abc123",
        "assertionId": "1b4e28ba-2fa1-4d3b-9c6e-0f2a7d9e5c11",
        "text": "Steady-state traffic is about 500 requests per second, with
 predictable spikes to 5,000 requests per second during weekday business hours.",
        "source": "USER",
        "createdAt": "2026-09-11T14:32:10.000000+00:00",
        "updatedAt": "2026-09-11T14:32:10.000000+00:00"
    }
}
```

You can optionally pass `--client-token` to make the request idempotent.

To list assertions

```
aws resiliencehubv2 list-assertions \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/checkout:abc123"
```

To list only the assertions generated by the assessment engine, filter by source:

```
aws resiliencehubv2 list-assertions \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/checkout:abc123"
 \
  --source AI_GENERATED
```

To update an assertion

```
aws resiliencehubv2 update-assertion \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/checkout:abc123"
 \
  --assertion-id "1b4e28ba-2fa1-4d3b-9c6e-0f2a7d9e5c11" \
  --text "Steady-state traffic is about 800 requests per second, with predictable
 spikes to 8,000 requests per second during weekday business hours."
```

If the assertion was AI-generated, the response shows its source changed to USER.

To delete an assertion

```
aws resiliencehubv2 delete-assertion \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/checkout:abc123"
 \
  --assertion-id "1b4e28ba-2fa1-4d3b-9c6e-0f2a7d9e5c11"
```

For the full request and response syntax, see CreateAssertion, ListAssertions, UpdateAssertion, and DeleteAssertion in the *Next generation Resilience Hub API Reference*.

Assertion limits

Each service can have up to 20 assertions. This limit is shared between AI-generated and user-created assertions and cannot be increased. If a service already has 20 assertions, `CreateAssertion` returns a `ServiceQuotaExceededException`. Delete assertions that no

longer apply to make room for new ones. Assertion text is limited to 1,000 characters. For more information, see Quotas and limits.

Service design files

You can attach architecture design documents stored in Amazon S3 to your service as input sources. The assessment engine reads these documents to understand your intended architecture and to identify gaps between the design and what is deployed. Supported formats include PDF, Markdown, and plain text. For more information, see Import design documents.

Running a failure mode assessment at the service level

You can run a failure mode assessment from the console or using the AWS CLI. The assessment runs asynchronously. Typical completion time is 5 to 15 minutes depending on service complexity.

Prerequisites

- The service's invoker role must be configured and accessible. For more information, see Required IAM permissions and roles.
- The service must have a valid discovered topology with a data flow relationship between at least two resources. For more information, see Minimum topology requirements.
- At least one resilience policy should be applied. Assessments without policies still run but produce fewer targeted findings.

Minimum topology requirements

To run a failure mode assessment, your service must have a topology with a data flow relationship between at least two resources.

A failure mode assessment analyzes how failures propagate through your architecture by examining the relationships between resources. Without a data flow relationship connecting at least two resources, there is no data flow to analyze. The assessment cannot produce meaningful findings.

Valid topology (assessment can run)

Any topology where at least two resources are connected by a data flow relationship is valid. For example:

- An Application Load Balancer routing traffic to an Amazon ECS cluster.

- A Lambda function reading from a DynamoDB table.
- An Amazon SQS queue feeding messages to a Lambda function.

Invalid topology (assessment cannot run)

A topology with only isolated resources and no data flow relationships between them is invalid. For example, a service that discovers a single Amazon S3 bucket or several unconnected Amazon EC2 instances has no data flow for the assessment to analyze.

To start a failure mode assessment (console)

1. Navigate to your service.
2. Choose **Failure mode guidance** and add any assertions about your service. For more information, see Failure mode guidance.
3. Choose **Run failure mode assessment**.
4. Wait for the assessment to complete (typically 5 to 15 minutes).

To start a failure mode assessment (CLI)

```
aws resiliencehubv2 start-failure-mode-assessment \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/checkout:abc123"
```

To check assessment status

```
aws resiliencehubv2 list-failure-mode-assessments \
  --service-arn "arn:aws:resiliencehub:..."
```

Status values progress as follows: `PENDING`, then `IN_PROGRESS`, then `SUCCESS` or `FAILED`.

During the assessment, Next generation Resilience Hub runs resource discovery in the background:

1. Next generation Resilience Hub assumes your invoker role.
2. Reads resources from your configured input sources (CloudFormation, tags, Terraform, or Amazon EKS).
3. Identifies parent-child relationships (for example, Auto Scaling group to Amazon EC2 instances).
4. Resilience Hub builds a topology of your service.

5. Builds a topology showing data flow and containment.

Once topology is complete, you can view it in the console:

- **Graph view** – Visual map of resources and connections.
- **Table view** – List of all discovered resources with metadata.
- **JSON export** – Download the full topology for external analysis.

Viewing failure mode findings and recommendations

After an assessment completes, review the findings and recommendations:

To view findings (console)

1. Navigate to your service and choose the **Assessment** tab.
2. Review findings sorted by severity.
3. Choose any finding to see detailed reasoning and recommendations.

To list findings (CLI)

```
aws resiliencehubv2 list-failure-mode-findings \
  --service-arn "arn:aws:resiliencehub:..."
```

Each finding includes the following information:

Field	Description	Example
Finding name	Brief description of the failure mode	"Single-AZ Amazon Relational Database Service database creates a critical single point of failure"
Description	Detailed explanation of the issue	Paragraph explaining why this is a risk
Reasoning	Why this matters for your specific architecture	Architecture-specific context for the finding

Field	Description	Example
Severity	Impact level	High, Medium, Low
Failure category	Type of failure	Shared fate, Excessive load, Excessive latency, Misconfiguration and bugs, Single points of failure
Policy component	Which policy requirements this relates to	Multi-AZ disaster recovery
Recommendations	Suggested mitigations	See recommendations table below

Each recommendation includes a mitigation, observability, and testing recommendation:

- **Mitigation** – An infrastructure or configuration change to address the failure mode.
- **Observability** – Monitoring or alerting improvements to detect the failure mode.
- **Testing** – A test to validate that the mitigation works as expected.

Managing findings

For each finding, you can reason about the failure mode in detail, implement the recommended fix, mark the finding as irrelevant if it does not apply to your use case, or use severity to prioritize what to address first.

Findings can have the following statuses:

- **Open** – Requires attention.
- **Resolved** – The issue has been fixed. The next assessment verifies the fix.
- **Irrelevant** – Suppressed by the customer. The finding stays suppressed across future assessments.

To mark a finding as resolved (CLI)

```
aws resiliencehubv2 update-failure-mode-finding \
  --service-arn "arn:aws:resiliencehub:..." \
  --finding-id "finding-123" \
```

```
  --status "RESOLVED"
```

To mark a finding as irrelevant (CLI)

```
aws resiliencehubv2 update-failure-mode-finding \
  --service-arn "arn:aws:resiliencehub:..." \
  --finding-id "finding-123" \
  --status "IRRELEVANT"
```

Assessment history and trends

Next generation Resilience Hub retains assessment history for 2 years, enabling you to:

- Track how your resilience posture improves over time.
- See which findings have been resolved versus remain open.
- Identify recurring issues across assessments.
- Generate compliance reports showing improvement trends.

To view assessment history (CLI)

```
aws resiliencehubv2 list-failure-mode-assessments \
  --service-arn "arn:aws:resiliencehub:..."
```

Pricing

Each service includes **2 failure mode assessments per month** in the base $15 per service per month fee for services with up to 150 resources.

For services with more than 150 resources, each of the 2 included assessments is charged at $0.10 per resource above the 150-resource threshold. Additional assessments beyond the 2 included per month are charged at the same per-resource rate, with a minimum of 50 resources per assessment.

Service size	Monthly cost for 2 assessments
100 resources	$15 (included)
150 resources	$15 (included)
200 resources	$15 + (50 x $0.10 x 2) = $25

Service size	Monthly cost for 2 assessments
500 resources	$15 + (350 x $0.10 x 2) = $85

Resilience testing

In the next generation of Resilience Hub there are four AWS recommended test templates, which have been developed to help you validate your services operate and recover as expected during failures. Three tests align to the AWS fault isolation boundaries (Availability Zones and Regions) and provide a consistent way to test your services' recovery capability to failures scoped to those boundaries. There is an additional test to help you validate your services can recover when there are failures to its dependencies. Each test is built on AWS Fault Injection Service (AWS FIS), which performs the underlying fault injection. Next generation Resilience Hub scopes the test to the resources in your service and evaluates the results against your chosen success criteria to determine a pass or fail outcome. You run resilience tests at the service level.

Topics

- How resilience testing works
- Configuring a test
- Available tests
- Test runs and report
- IAM execution roles for resilience testing

How resilience testing works

Resilience testing uses three core concepts:

- **Test template** – A pre-configured, recommended test that defines which resilience capability to validate. A test template specifies the actions to perform, and declares the parameters it accepts.
- **Test** – A test you create by configuring a recommended test template for your service, including your success criteria, stop conditions, and other settings. Each service has one test per template. For example, an Availability Zone: recovery test configured for your checkout service with a 30-minute RTO (Recovery Time Objective).
- **Test run** – A single execution of a test. Each test run scopes to the current resources in your service and produces a pass or fail outcome, along with results and a report. You can have

multiple runs of the same test. For example, running your Availability Zone: recovery test monthly to test recovery within your 30-minute RTO.

When you run a resilience test, the next generation of Resilience Hub performs the following:

1. **Scopes the test** – Automatically includes all resources discovered in your service that apply to the test.
2. **Configures the test** – Sets up the test with the configuration you provide, including success alarms, stop conditions, and log destinations.
3. **Injects faults** – Runs AWS FIS experiments to inject the faults defined by the test for the period of time specified.
4. **Evaluates results** – Determines a pass or fail outcome based on whether your service met your success criteria during the test. You can review test results in the console or programmatically. A test report is generated for each run.

Resilience tests are charged based on AWS FIS action-minute pricing. For details, see AWS Fault Injection Service pricing.

Configuring a test

When you create a test from a template, you configure the following settings. Settings are saved and reused for future runs – you can edit them at any time.

Test duration

The length of time the test actions run. For recovery tests (**Availability Zone: recovery** and **Multi-Region: recovery**), the default duration is your RTO plus 30 minutes, giving your service time to recover and confirming that it sustains. For sustained tests, the default is 30 minutes for **Dependency validation** or 3 hours for **Multi-Region: isolation**. These defaults apply in the console; when using the API, you provide the duration explicitly.

Fault scope

Defines where faults are injected – the Availability Zone or Regions to impair. Each test has its own parameters. For details, see Available tests.

Test actions

Read-only. Defined by the test template. Shows the fault actions and their target resource types. If no resources match an action's target type, that action is skipped.

Dependencies to block

Dependencies to block during the test. Select from discovered dependencies (if dependency discovery is enabled) or enter dependencies manually by DNS domain name. Required for some tests, optional for others. For defaults and requirements, see Available tests.

Test alarms

Success alarms (required) – CloudWatch alarms that determine success criteria pass or fail. At least one is required. If you select multiple, all must pass for the test to pass. Choose alarms that measure your service's overall health.

How the test passes:

- **Availability Zone: recovery** – all success alarms must return to OK state within your Multi-AZ RTO and remain there until the test actions end.
- **Multi-Region: recovery** – all success alarms must return to OK state within your Multi-Region RTO and remain there until the test actions end.
- **Dependency validation** – all success alarms remain in OK state until the test actions end.
- **Multi-Region: isolation** – all success alarms remain in OK state until the test actions end.

Additional alarms (optional) – For observability only. Shown in the results and report but do not affect the pass or fail outcome.

Region switch plan (**Multi-Region: recovery** only, optional)

Attaching an AWS Application Recovery Controller (ARC) Region Switch plan lets the next generation of Resilience Hub include the failover timeline in your test results and report. Optional; leave empty if you use manual failover or a custom automation.

Additional settings (optional)

Stop conditions

CloudWatch alarms that automatically stop the test if they breach their threshold. Use stop conditions as guardrails to protect your workloads during testing. For example, create a CloudWatch alarm on a critical business metric (such as latency or error rate) that stops the test if it exceeds an unacceptable threshold. Stop condition alarms must be in the account

where the test runs and the Region where the faults are injected. For more information, see Stop conditions for AWS FIS.

Reports

A report is automatically generated after each test run when a report destination (Amazon S3 bucket) is configured on your service. Reports include the test configuration, alarm status, timeline, and pass or fail results. Test reports are included at no additional cost when you use the next generation of Resilience Hub resilience testing. To configure a report destination, see your service's reporting settings.

Logs

Send detailed test logs to Amazon S3, CloudWatch Logs, or both. Logs capture timestamped events, including action start and end, target resolution, and errors. Use logs to debug failed tests or to understand the sequence of fault injection. Logging requires additional permissions on your test execution role. For more information, see Experiment logging for AWS FIS.

Test permissions

An IAM execution role is required. Permissions vary based on the test type, fault actions, and options selected. For multi-account tests, use the same-named role in each account. For more information, see IAM execution roles for resilience testing.

Tests that inject faults into Amazon EKS pods also need a Kubernetes service account and cluster access for the execution role. You set these up in the **(Optional) Resilience testing** parts of the Amazon EKS permissions steps. For more information, see Required IAM permissions and roles.

Available tests

Next generation Resilience Hub provides the following pre-configured test templates:

- **Availability Zone: recovery** – Validate your service can detect and recover from an Availability Zone impairment.
- **Dependency validation** – Validate how your service responds when its dependencies within the Region are impaired.
- **Multi-Region: isolation** – Validate your service can operate independently from a Region when connectivity to another Region is unavailable.

- **Multi-Region: recovery** – Validate your service can recover and serve clients from another Region within your recovery objectives when a Region is impaired.

Each test template declares the parameters it accepts, including whether each parameter is required and any default values.

To list test templates (CLI)

```
aws resiliencehubv2 list-test-templates --region region
```

To view a test template (CLI)

Use `get-test-template` to view the full definition of a test template. The response includes:

- **actions** – The operations the test performs, each targeting a specific resource type.
- **parameters** – The inputs you provide when you create a test from the template.

Each action includes the following:

- **actionId** – A unique identifier for the action.
- **description** – A description of what the action does.
- **resourceType** – The type of resource the action targets.

Each parameter declares the following:

- **name** – The parameter name you supply when you create a test.
- **description** – A description of what the parameter controls.
- **type** – The value type: `STRING`, `STRING_LIST`, or `INTEGER`.
- **required** – Whether you must provide a value for the parameter.
- **defaultValue** – The value used when you don't provide one. Applies to optional parameters.
- **maxValues** – The maximum number of values you can provide for the parameter.

```
aws resiliencehubv2 get-test-template \
  --test-template-arn "arn:aws:resiliencehub:region:aws:test-template/aws-az-
recovery:rtaz001" \
```

```
  --region region
```

Provide values for these parameters with the `--parameters` option when you create a test. For an example of creating a test, see Getting started with Next generation Resilience Hub.

Topics

- Availability Zone: recovery
- Dependency validation
- Multi-Region: isolation
- Multi-Region: recovery

Availability Zone: recovery

The **Availability Zone: recovery** test injects the symptoms of a power interruption in a single Availability Zone, affecting compute, networking, storage, and database resources within that AZ. It can help you validate that your service detects the impaired AZ, continues operating in the healthy AZs, and returns to a healthy state within your defined RTO. In addition, this test can help you surface single-AZ dependencies you may not be aware of.

What makes this test unique

- Targets a single Availability Zone that you choose, focusing on zonal, in-Region resilience.
- This is a recovery test – your service must recover within your RTO.

How to pass this test

- This is a recovery test. The RTO countdown starts when test actions begin. The test passes if all success alarms return to OK state within your RTO and remain there until the test actions end.

Things to think about

- Choose success alarms that measure regional service health (for example, regional error rate, latency) – avoid per-AZ alarms since the impaired AZ is expected to be unhealthy.
- Most meaningful for multi-AZ architectures where traffic can shift to healthy AZs. Can also be run against single-AZ services.

- If you have AWS Application Recovery Controller (ARC) zonal autoshift enabled on your resources, the test exercises that shift automatically. If autoshift is not configured, the action is skipped.

Key test parameters

- **Availability Zone** – Choose the AZ to impair. Select an AZ where your service has resources deployed.
- **Duration** – The length of time the test actions run. It takes a few additional minutes afterward to collect final results before the test ends. Defaults to your RTO from your service policy plus 30 minutes when you first create the test. Set it longer than your RTO to validate that recovery is sustained. The minimum duration is 10 minutes.

Actions

This test runs the following AWS FIS actions to impair the Availability Zone that you select. If your service has no resources matching an action's target type, that action is skipped. For details about each action, see the AWS FIS actions reference.

Action	Description
`aws:ec2:stop-instances`	Stops instances in the impaired AZ and restarts them 5 minutes before the test actions end.
`aws:ec2:api-insufficient-instance-capacity-error`	Blocks new instance launches in the impaired AZ.
`aws:ec2:asg-insufficient-instance-capacity-error`	Prevents Auto Scaling from provisioning capacity in the AZ.
`aws:network:disrupt-connectivity`	Blocks traffic entering and leaving the subnet, and blocks access to Amazon S3 Express One Zone directory buckets if present.
`aws:rds:failover-db-cluster`	Fails over the cluster if the writer is in the impaired AZ.

Action	Description
`aws:elasticache:replicationgroup-interrupt-az-power`	Terminates cache nodes in the impaired AZ without replacement for the duration.
`aws:ebs:pause-volume-io`	After instances restart, pauses I/O for 5 minutes on half of the Amazon EBS volumes in the impaired AZ (at least one). Only volumes that persist after instance termination are targeted.
`aws:arc:start-zonal-autoshift`	Shifts traffic to healthy AZs.

To see this test's parameters and their default values, use `get-test-template`.

Dependency validation

The **Dependency validation** test blocks dependencies within the Region to see how your service behaves when they are unavailable. It can help you validate that your service remains healthy when soft dependencies are impaired, and surface dependencies you may not have known were hard. You can also include known hard dependencies to understand their impact.

What makes this test unique

- Focuses on in-Region dependency failures.
- You choose which dependencies to target.
- If you have enabled dependency discovery in the next generation of Resilience Hub, you can pick from discovered dependencies. If not, you can enter DNS endpoints manually.

How to pass this test

- This is a sustained test. The test passes if all success alarms remain in OK state until the test actions end. Use this to validate soft dependencies are truly soft – if your alarms breach, the dependency may actually be hard.

Things to think about

- Start with a single soft dependency and build up – validate one at a time before blocking many.
- If you include a hard dependency, expect your alarms to breach and the test to fail – hard dependencies cause significant impact when blocked. This is useful to confirm that a dependency is truly hard.
- Choose success alarms that measure overall service health, not the health of the dependency itself.
- Ensure your dependencies are actively used during the test (traffic is flowing to them) – this validates the block is having an effect. Consider adding alarms or metrics that track dependency usage (for example, request count or connection errors) to verify that the dependency is being exercised during the test.
- Dependencies must be resolvable DNS endpoints – if the DNS doesn't resolve, the action will fail.
- Blocking dependencies that trigger health check failures may cause compute (for example, Amazon ECS tasks) to be replaced. The packet loss action does not re-apply to replacement tasks and may report as failed.

Key test parameters

- **Duration** – The length of time the test actions run. It takes a few additional minutes afterward to collect final results before the test ends. Default is 30 minutes when you first create the test.
- **Dependencies to block** – By default, the discovered soft dependency with the highest query volume is pre-selected. If no dependencies have been classified as soft, none are selected. You can adjust to target individual dependencies or specific groups, or manually add dependencies by DNS domain name. Additional dependencies added here are only used for this test and will not be saved to the service's dependency discovery. These defaults apply in the console; when using the API, you provide the dependencies explicitly.

Actions

This test runs the following AWS FIS actions to drop traffic to the dependencies that you select. Actions inject 100% packet loss on Amazon EC2 instances, Amazon ECS tasks (Amazon EC2 and Fargate), and Amazon EKS pods (Amazon EC2). If your service has no resources matching an action's target type, that action is skipped.

Note

The actions used to block dependencies require additional setup: SSM Agent installed on Amazon EC2 instances, an SSM Agent container in your Amazon ECS task definition, or a Kubernetes service account for Amazon EKS pods. For Amazon EKS pods, complete the **(Optional) Resilience testing** parts of the Amazon EKS permissions steps. For more information, see Required IAM permissions and roles.

Action	Description
`aws:ssm:send-command`	Drops traffic from Amazon EC2 instances to the selected dependencies.
`aws:ecs:task-network-packet-loss`	Drops traffic from Amazon ECS tasks to the selected dependencies.
`aws:eks:pod-network-packet-loss`	Drops traffic from Amazon EKS pods to the selected dependencies.

To see this test's parameters and their default values, use `get-test-template`.

Multi-Region: isolation

The **Multi-Region: isolation** test blocks connectivity between two Regions, including network traffic, cross-Region data replication, and selected dependencies. It can help you validate that your service operates independently when resources in another Region are not accessible, and surface cross-Region dependencies that could impact availability.

What makes this test unique

- Focuses on Region isolation (loss of cross-Region connectivity) rather than recovery to another Region.
- Blocks network traffic, pauses cross-Region replication (Amazon S3, DynamoDB, MemoryDB), and optionally blocks dependencies between the two Regions.
- Validates that the Region can function on its own.

How to pass this test

- This is a sustained test. The test passes if all success alarms remain in OK state until the test actions end.

Things to think about

- Choose success alarms that measure the isolated Region's health – validate it continues serving traffic independently.
- This test applies to both active/active and active/passive architectures.
- A key use case: validate your recovery Region can operate standalone. For example, if your primary is `us-east-1` and secondary is `us-west-2`, use this to validate `us-west-2` operates independently when isolated from `us-east-1`.
- This test is also a good precursor to **Multi-Region: recovery** – validate independence before testing full failover. The test can help discover cross-Region dependencies that are critical.
- Ensure your dependencies are actively used during the test (traffic is flowing to them) – this validates the block is having an effect. Consider adding alarms or metrics that track dependency usage (for example, request count or connection errors) to verify that the dependency is being exercised during the test.
- Dependencies must be resolvable DNS endpoints.
- Blocking dependencies that trigger health check failures may cause compute (for example, Amazon ECS tasks) to be replaced. The packet loss action does not re-apply to replacement tasks and may report as failed.

Key test parameters

- **Isolated Region** – The Region you are testing for independent operation.
- **Destination Region** – The Region being blocked (connectivity cut to this Region).
- **Duration** – The length of time the test actions run. It takes a few additional minutes afterward to collect final results before the test ends. Default is 3 hours when you first create the test.
- **Dependencies to block** (optional) – If dependency discovery is enabled, the list shows dependencies from the isolated Region to the destination Region. You can select from this list or add manually by DNS domain name. No dependencies are selected by default. Other actions in this test (network connectivity, replication pausing) run regardless. Additional dependencies added here are only used for this test and will not be saved to the service's dependency

discovery. These defaults apply in the console; when using the API, you provide the dependencies explicitly.

Actions

This test runs the following AWS FIS actions to drop traffic to the dependencies that you select. Actions inject 100% packet loss on Amazon EC2 instances, Amazon ECS tasks (Amazon EC2 and Fargate), and Amazon EKS pods (Amazon EC2). If your service has no resources matching an action's target type, that action is skipped.

> **Note**
>
> The actions used to block dependencies require additional setup: SSM Agent installed on Amazon EC2 instances, an SSM Agent container in your Amazon ECS task definition, or a Kubernetes service account for Amazon EKS pods. For Amazon EKS pods, complete the **(Optional) Resilience testing** parts of the Amazon EKS permissions steps. For more information, see Required IAM permissions and roles.

Action	Description
`aws:network:transit-gateway-disrupt-cross-region-connectivity`	Blocks cross-Region traffic via Transit Gateway peering to the destination Region.
`aws:network:route-table-disrupt-cross-region-connectivity`	Blocks cross-Region traffic from subnets to the destination Region.
`aws:network:disrupt-vpc-endpoint`	Blocks traffic to cross-Region VPC endpoints to the destination Region.
`aws:s3:bucket-pause-replication`	Pauses Amazon S3 cross-Region replication to the destination Region.
`aws:dynamodb:global-table-pause-replication`	Pauses DynamoDB global table replication.

Action	Description
`aws:memorydb:multi-region-cluster-pause-replication`	Pauses MemoryDB multi-Region cluster replication.
`aws:ssm:send-command`	Drops traffic from Amazon EC2 instances to the selected dependencies.
`aws:ecs:task-network-packet-loss`	Drops traffic from Amazon ECS tasks to the selected dependencies.
`aws:eks:pod-network-packet-loss`	Drops traffic from Amazon EKS pods to the selected dependencies.

To see this test's parameters and their default values, use `get-test-template`.

Multi-Region: recovery

The **Multi-Region: recovery** test introduces failures into dependencies in one Region to validate that your service can recover and serve clients from a recovery Region within your recovery objectives. This test applies to both active/active and active/passive architectures. For example, you can initiate your recovery procedure and confirm your service recovers within your defined Recovery Time Objective (RTO) in the recovery Region.

What makes this test unique

- Validates recovery to another Region against your recovery objectives, not just in-Region resilience.
- You choose which dependencies to impair in the primary Region, so you can practice detection and recovery.
- The test integrates with ARC Region switch so you can track plan execution details in the test report.

How to pass this test

- This is a recovery test. The RTO countdown starts when test actions begin. The test passes if all success alarms return to 0K state within your Multi-Region RTO and remain there until the test actions end. Your service must have a resilience policy with a Multi-Region RTO defined.

Things to think about

- Choose dependencies in the impaired Region that are significant enough to trigger your recovery procedure. Pick hard dependencies or DNS endpoints that, if blocked, would meaningfully impair that Region.
- The test injects faults in the impaired Region – you perform the recovery action yourself (for example, triggering failover/ARC Region switch plan). The test watches whether your recovery Region comes healthy within your RTO by evaluating alarm state.
- Choose success alarms in the recovery Region that validate it's serving traffic – or use global/application-level alarms that reflect overall customer experience.
- Consider setting duration longer than your RTO to validate recovery is sustained.
- Ensure your dependencies are actively used during the test (traffic is flowing to them) – this validates the block is having an effect. Consider adding alarms or metrics that track dependency usage (for example, request count or connection errors) to verify that the dependency is being exercised during the test.
- Dependencies must be resolvable DNS endpoints.
- Blocking dependencies that trigger health check failures may cause compute (for example, Amazon ECS tasks) to be replaced. The packet loss action does not re-apply to replacement tasks and may report as failed.

Key test parameters

- **Impaired Region** – The Region where faults are injected.
- **Recovery Region** – The Region where you expect your service to recover to.
- **Duration** – The length of time the test actions run. It takes a few additional minutes afterward to collect final results before the test ends. Defaults to your Multi-Region RTO from your service policy plus 30 minutes when you first create the test.
- **Dependencies to block** – Choose dependencies that, if blocked, would significantly impair your service and help validate failover to another Region. By default, the discovered hard dependency

with the highest query volume is pre-selected. If no dependencies have been classified as hard, none are selected. You can adjust or manually add dependencies by DNS domain name. Additional dependencies added here are only used for this test and will not be saved to the service's dependency discovery. These defaults apply in the console; when using the API, you provide the dependencies explicitly.

- **Region Switch plan** (optional) – Attaching an ARC Region Switch plan lets the next generation of Resilience Hub include the actual failover timeline in your test results and report. If you use manual failover or a custom automation, leave this empty.

Actions

This test runs the following AWS FIS actions to drop traffic to the dependencies that you select. Actions inject 100% packet loss on Amazon EC2 instances, Amazon ECS tasks (Amazon EC2 and Fargate), and Amazon EKS pods (Amazon EC2). If your service has no resources matching an action's target type, that action is skipped.

> **Note**
>
> The actions used to block dependencies require additional setup: SSM Agent installed on Amazon EC2 instances, an SSM Agent container in your Amazon ECS task definition, or a Kubernetes service account for Amazon EKS pods. For Amazon EKS pods, complete the **(Optional) Resilience testing** parts of the Amazon EKS permissions steps. For more information, see Required IAM permissions and roles.

Action	Description
`aws:ssm:send-command`	Drops traffic from Amazon EC2 instances to the selected dependencies.
`aws:ecs:task-network-packet-loss`	Drops traffic from Amazon ECS tasks to the selected dependencies.
`aws:eks:pod-network-packet-loss`	Drops traffic from Amazon EKS pods to the selected dependencies.

To see this test's parameters and their default values, use `get-test-template`.

Test runs and report

Test runs

Each time you start a test, a test run is created. You can view test runs in multiple places:

- **Service level**: Navigate to your service → Testing tab → test runs at the bottom. Shows all runs for that service.
- **Test runs page** (left nav): Shows test runs across all your services. Filter by service, test name, or status.
- **Dashboard**: The test results chart shows pass/fail distribution across your organization.

Each test run includes the test status, timing information, success alarm results, a timeline of actions, and any dependencies blocked or log events captured during the run. You can drill into a test run to see full details including the AWS FIS experiment ID and individual action outcomes.

You can have multiple runs of the same test. For example, running your **Availability Zone: recovery** test monthly to track resilience improvements over time. You can also list test runs and retrieve test run details programmatically using the `ListTestRuns` and `GetTestRun` API operations. To retrieve the Amazon CloudWatch alarm state-change timeline observed during a run and the dependencies the run blocked, use the `ListTestRunSourceEvents` and `ListTestRunDependencies` API operations. See the API reference.

Test reports

A test report is auto-generated after each test run completes, if a report destination (Amazon S3 bucket) is configured on your service. If no destination is configured, no report is generated.

Reports capture the test configuration, success criteria results, a visual timeline of how the service responded, and the overall pass/fail outcome.

Where to find reports:

- **Reports page** (left nav): Shows all reports (assessment and test) across your services.
- **Test run details**: Download directly from the test run detail page.

Test reports are included at no additional cost when using the next generation of Resilience Hub resilience testing.

You can also retrieve test run results and report metadata programmatically using the `GetTestRun` and `ListTestRunSources` API operations. See the API reference.

IAM execution roles for resilience testing

To run a resilience test, you provide an IAM *execution role* that AWS Fault Injection Service (AWS FIS) assumes to inject faults into your resources. You set this role on the test, and Resilience Hub uses it when you start a test run.

When you create a test in the console, Resilience Hub can create this role for you. Use the policies in this topic when you create or customize the role yourself, for example with the AWS CLI.

The permissions a role needs depend on the test template, because each template runs a different set of AWS FIS actions. This topic provides a permissions policy for each of the four test templates. For more information about the templates, see Available tests.

Resilience Hub supports two account models:

- **Single-account** – The test targets resources in the same account that runs the experiment. You create one execution role.
- **Multi-account** – The test targets resources in other accounts. You create an orchestrator role in the account that runs the experiment, and a target role in each account that contains targeted resources.

We recommend that you follow the standard security practice of granting least privilege. The following policies scope each permission to the resource type that the action targets. Resilience testing can't scope permissions to individual resource IDs, because the next generation of Resilience Hub resolves the targeted resources at run time.

Topics

- Single-account execution role
- Multi-account execution roles

Single-account execution role

For a single-account test, create one execution role with a trust policy that allows AWS FIS to assume it and a permissions policy for the test template that you use.

Trust policy

The execution role must trust the AWS FIS service. The trust policy is the same for every test template. The `aws:SourceAccount` and `aws:SourceArn` conditions protect against the confused deputy problem, so that only experiments owned by your account can assume the role.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Sid": "FISTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "fis.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "account-id"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:fis:*:account-id:experiment/*"
        }
      }
    }
  ]
}
```

Permissions policy

Attach the permissions policy for the test template that your test uses.

Topics

- Availability Zone: recovery
- Dependency validation
- Multi-Region: isolation
- Multi-Region: recovery
- Create the role and attach it to your test

Availability Zone: recovery

The **Availability Zone: recovery** template runs AWS FIS actions against the following services: Amazon EC2, Amazon EBS, Amazon EC2 Auto Scaling, Amazon ElastiCache, Amazon RDS, network ACLs, and AWS Application Recovery Controller zonal shift. Attach the following permissions policy to the execution role.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Sid": "EC2StopAndStartInstances",
      "Effect": "Allow",
      "Action": [
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": "arn:aws:ec2:*:account-id:instance/*"
    },
    {
      "Sid": "EC2DescribeInstances",
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    },
    {
      "Sid": "EC2EncryptedVolumesKmsGrant",
      "Effect": "Allow",
      "Action": "kms:CreateGrant",
      "Resource": "arn:aws:kms:*:account-id:key/*",
      "Condition": {
        "StringLike": {
          "kms:ViaService": "ec2.*.amazonaws.com"
        },
        "Bool": {
          "kms:GrantIsForAWSResource": "true"
        },
        "ForAllValues:StringEquals": {
          "kms:GrantOperations": ["Decrypt", "CreateGrant"]
        },
        "Null": {
          "kms:GrantOperations": "false"
```

```
        },
        "StringEquals": {
          "kms:GrantConstraintType": "EncryptionContextSubset"
        },
        "ForAnyValue:StringEquals": {
          "kms:EncryptionContextKeys": "aws:ebs:id"
        }
      }
    },
    {
      "Sid": "EC2InjectApiError",
      "Effect": "Allow",
      "Action": "ec2:InjectApiError",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "ec2:FisActionId": [
            "aws:ec2:api-insufficient-instance-capacity-error",
            "aws:ec2:asg-insufficient-instance-capacity-error"
          ]
        }
      }
    },
    {
      "Sid": "AutoScalingDescribe",
      "Effect": "Allow",
      "Action": [
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeTags"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ElastiCacheInterruptAzPower",
      "Effect": "Allow",
      "Action": [
        "elasticache:InterruptClusterAzPower",
        "elasticache:DescribeReplicationGroups"
      ],
      "Resource": "arn:aws:elasticache:*:account-id:replicationgroup:*"
    },
    {
      "Sid": "EBSPauseVolumeIO",
      "Effect": "Allow",
```

```
      "Action": "ec2:PauseVolumeIO",
      "Resource": "arn:aws:ec2:*:account-id:volume/*"
    },
    {
      "Sid": "EBSDescribeVolumes",
      "Effect": "Allow",
      "Action": "ec2:DescribeVolumes",
      "Resource": "*"
    },
    {
      "Sid": "NetworkTagManagedNacl",
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:*:account-id:network-acl/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkAcl",
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "NetworkCreateManagedNacl",
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkAcl",
      "Resource": "arn:aws:ec2:*:account-id:network-acl/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "NetworkCreateNaclOnVpc",
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkAcl",
      "Resource": "arn:aws:ec2:*:account-id:vpc/*"
    },
    {
      "Sid": "NetworkModifyManagedNacl",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkAclEntry",
        "ec2:DeleteNetworkAcl"
```

```
      ],
      "Resource": "arn:aws:ec2:*:account-id:network-acl/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "NetworkDescribe",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeNetworkAcls",
        "ec2:DescribeManagedPrefixLists"
      ],
      "Resource": "*"
    },
    {
      "Sid": "NetworkReplaceNaclAssociation",
      "Effect": "Allow",
      "Action": "ec2:ReplaceNetworkAclAssociation",
      "Resource": [
        "arn:aws:ec2:*:account-id:subnet/*",
        "arn:aws:ec2:*:account-id:network-acl/*"
      ]
    },
    {
      "Sid": "NetworkPrefixListEntries",
      "Effect": "Allow",
      "Action": "ec2:GetManagedPrefixListEntries",
      "Resource": "arn:aws:ec2:*:account-id:prefix-list/*"
    },
    {
      "Sid": "RDSFailoverCluster",
      "Effect": "Allow",
      "Action": [
        "rds:FailoverDBCluster",
        "rds:DescribeDBClusters"
      ],
      "Resource": "arn:aws:rds:*:account-id:cluster:*"
    },
    {
```

```
      "Sid": "RDSDescribeForTargetResolution",
      "Effect": "Allow",
      "Action": "rds:DescribeDBInstances",
      "Resource": "arn:aws:rds:*:account-id:db:*"
    },
    {
      "Sid": "ARCZonalShiftManagedElb",
      "Effect": "Allow",
      "Action": [
        "arc-zonal-shift:StartZonalShift",
        "arc-zonal-shift:GetManagedResource",
        "arc-zonal-shift:UpdateZonalShift",
        "arc-zonal-shift:CancelZonalShift"
      ],
      "Resource": [
        "arn:aws:elasticloadbalancing:*:account-id:loadbalancer/app/*",
        "arn:aws:elasticloadbalancing:*:account-id:loadbalancer/net/*"
      ]
    },
    {
      "Sid": "ARCZonalShiftManagedAsgEks",
      "Effect": "Allow",
      "Action": [
        "arc-zonal-shift:StartZonalShift",
        "arc-zonal-shift:GetManagedResource",
        "arc-zonal-shift:UpdateZonalShift",
        "arc-zonal-shift:CancelZonalShift"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "arc-zonal-shift:ResourceIdentifier": [
            "arn:aws:autoscaling:*:account-id:autoScalingGroup:*",
            "arn:aws:eks:*:account-id:cluster/*"
          ]
        }
      }
    },
    {
      "Sid": "ARCZonalShiftList",
      "Effect": "Allow",
      "Action": "arc-zonal-shift:ListManagedResources",
      "Resource": "*"
    },
```

```
    {
      "Sid": "TargetResolutionByTags",
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
    },
    {
      "Sid": "FISExperimentLogging",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs:DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": "*"
    },
    {
      "Sid": "FISExperimentLogGroupAccess",
      "Effect": "Allow",
      "Action": [
        "logs:DescribeResourcePolicies",
        "logs:PutResourcePolicy",
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

Dependency validation

The **Dependency validation** template runs AWS FIS actions that block in-Region dependency traffic from Amazon EC2, Amazon ECS, and Amazon EKS workloads. Attach the following permissions policy to the execution role.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
```

```
      "Sid": "SsmSendCommandOnDocuments",
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ssm:*::document/AWSFIS-Run-Network-Packet-Loss-Sources",
        "arn:aws:ssm:*::document/AWSFIS-Run-Network-Packet-Loss-ECS"
      ]
    },
    {
      "Sid": "SsmSendCommandOnInstances",
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ec2:*:account-id:instance/*",
        "arn:aws:ssm:*:account-id:managed-instance/*",
        "arn:aws:ecs:*:account-id:task/*/*"
      ]
    },
    {
      "Sid": "SsmListAndCancelCommands",
      "Effect": "Allow",
      "Action": [
        "ssm:ListCommands",
        "ssm:CancelCommand"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Ec2DescribeForTargetResolution",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeSubnets"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EcsDescribeForTargetResolution",
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeTasks",
        "ecs:DescribeContainerInstances",
        "ecs:ListTasks"
      ],
```

```
      "Resource": [
        "arn:aws:ecs:*:account-id:task/*/*",
        "arn:aws:ecs:*:account-id:container-instance/*/*",
        "arn:aws:ecs:*:account-id:cluster/*"
      ]
    },
    {
      "Sid": "EksDescribeCluster",
      "Effect": "Allow",
      "Action": "eks:DescribeCluster",
      "Resource": "arn:aws:eks:*:account-id:cluster/*"
    },
    {
      "Sid": "TargetResolutionByTags",
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
    },
    {
      "Sid": "FISExperimentLogging",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs:DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": "*"
    },
    {
      "Sid": "FISExperimentLogGroupAccess",
      "Effect": "Allow",
      "Action": [
        "logs:DescribeResourcePolicies",
        "logs:PutResourcePolicy",
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

Multi-Region: isolation

The **Multi-Region: isolation** template runs AWS FIS actions that disrupt cross-Region connectivity (route tables, transit gateways, and VPC endpoints) and pause cross-Region replication for Amazon S3, Amazon DynamoDB, and Amazon MemoryDB. Attach the following permissions policy to the execution role.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Sid": "RouteTableDisruptCrossRegion",
      "Effect": "Allow",
      "Action": [
        "ec2:AssociateRouteTable",
        "ec2:DisassociateRouteTable",
        "ec2:ReplaceRouteTableAssociation",
        "ec2:CreateRoute"
      ],
      "Resource": [
        "arn:aws:ec2:*:account-id:route-table/*",
        "arn:aws:ec2:*:account-id:subnet/*"
      ]
    },
    {
      "Sid": "RouteTableManagedCreate",
      "Effect": "Allow",
      "Action": "ec2:CreateRouteTable",
      "Resource": "arn:aws:ec2:*:account-id:route-table/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "RouteTableManagedCreateOnVpc",
      "Effect": "Allow",
      "Action": "ec2:CreateRouteTable",
      "Resource": "arn:aws:ec2:*:account-id:vpc/*"
    },
    {
```

```
      "Sid": "RouteTableManagedDelete",
      "Effect": "Allow",
      "Action": "ec2:DeleteRouteTable",
      "Resource": [
        "arn:aws:ec2:*:account-id:route-table/*",
        "arn:aws:ec2:*:account-id:vpc/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "PrefixListManagedCreate",
      "Effect": "Allow",
      "Action": "ec2:CreateManagedPrefixList",
      "Resource": "arn:aws:ec2:*:account-id:prefix-list/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "PrefixListManagedModifyDelete",
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteManagedPrefixList",
        "ec2:ModifyManagedPrefixList"
      ],
      "Resource": "arn:aws:ec2:*:account-id:prefix-list/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "NetworkInterfaceManagedCreate",
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkInterface",
      "Resource": "arn:aws:ec2:*:account-id:network-interface/*",
      "Condition": {
```

```
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "NetworkInterfaceManagedCreateOnSubnetAndSg",
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkInterface",
      "Resource": [
        "arn:aws:ec2:*:account-id:subnet/*",
        "arn:aws:ec2:*:account-id:security-group/*"
      ]
    },
    {
      "Sid": "NetworkInterfaceManagedDelete",
      "Effect": "Allow",
      "Action": "ec2:DeleteNetworkInterface",
      "Resource": "arn:aws:ec2:*:account-id:network-interface/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "FISCreateTags",
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": [
        "arn:aws:ec2:*:account-id:route-table/*",
        "arn:aws:ec2:*:account-id:prefix-list/*",
        "arn:aws:ec2:*:account-id:network-interface/*",
        "arn:aws:ec2:*:account-id:security-group/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "Ec2NetworkDescribeAndRead",
      "Effect": "Allow",
```

```
      "Action": [
        "ec2:DescribeRouteTables",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcPeeringConnections",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeManagedPrefixLists",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeSecurityGroups",
        "ec2:GetManagedPrefixListEntries",
        "ec2:DescribeTransitGateways",
        "ec2:DescribeTransitGatewayAttachments",
        "ec2:DescribeTransitGatewayPeeringAttachments",
        "ec2:DescribeInstances"
      ],
      "Resource": "*"
    },
    {
      "Sid": "TgwDisruptCrossRegion",
      "Effect": "Allow",
      "Action": [
        "ec2:AssociateTransitGatewayRouteTable",
        "ec2:DisassociateTransitGatewayRouteTable"
      ],
      "Resource": [
        "arn:aws:ec2:*:account-id:transit-gateway-route-table/*",
        "arn:aws:ec2:*:account-id:transit-gateway-attachment/*"
      ]
    },
    {
      "Sid": "VpcEndpointDisrupt",
      "Effect": "Allow",
      "Action": "ec2:ModifyVpcEndpoint",
      "Resource": [
        "arn:aws:ec2:*:account-id:vpc-endpoint/*",
        "arn:aws:ec2:*:account-id:security-group/*"
      ]
    },
    {
      "Sid": "VpcEndpointSecurityGroupManaged",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateSecurityGroup",
        "ec2:DeleteSecurityGroup",
```

```
        "ec2:RevokeSecurityGroupEgress"
      ],
      "Resource": [
        "arn:aws:ec2:*:account-id:security-group/*",
        "arn:aws:ec2:*:account-id:vpc/*"
      ]
    },
    {
      "Sid": "SsmSendCommandOnDocuments",
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ssm:*::document/AWSFIS-Run-Network-Packet-Loss-Sources",
        "arn:aws:ssm:*::document/AWSFIS-Run-Network-Packet-Loss-ECS"
      ]
    },
    {
      "Sid": "SsmSendCommandOnInstances",
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ec2:*:account-id:instance/*",
        "arn:aws:ssm:*:account-id:managed-instance/*",
        "arn:aws:ecs:*:account-id:task/*/*"
      ]
    },
    {
      "Sid": "SsmListAndCancelCommands",
      "Effect": "Allow",
      "Action": [
        "ssm:ListCommands",
        "ssm:CancelCommand"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EcsDescribeForTargetResolution",
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeTasks",
        "ecs:DescribeContainerInstances",
        "ecs:ListTasks"
      ],
      "Resource": [
```

```
        "arn:aws:ecs:*:account-id:task/*/*",
        "arn:aws:ecs:*:account-id:container-instance/*/*",
        "arn:aws:ecs:*:account-id:cluster/*"
      ]
    },
    {
      "Sid": "EksDescribeCluster",
      "Effect": "Allow",
      "Action": "eks:DescribeCluster",
      "Resource": "arn:aws:eks:*:account-id:cluster/*"
    },
    {
      "Sid": "DynamoDbGlobalTablePauseReplication",
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutResourcePolicy",
        "dynamodb:GetResourcePolicy",
        "dynamodb:DeleteResourcePolicy",
        "dynamodb:DescribeTable",
        "dynamodb:InjectError"
      ],
      "Resource": "arn:aws:dynamodb:*:account-id:table/*"
    },
    {
      "Sid": "S3PauseReplicationConfiguration",
      "Effect": "Allow",
      "Action": [
        "s3:PutReplicationConfiguration",
        "s3:GetReplicationConfiguration"
      ],
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "BoolIfExists": {
          "s3:IsReplicationPauseRequest": "true"
        },
        "StringEquals": {
          "aws:ResourceAccount": "account-id"
        }
      }
    },
    {
      "Sid": "S3PauseReplication",
      "Effect": "Allow",
      "Action": "s3:PauseReplication",
```

```
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "account-id"
        }
      }
    },
    {
      "Sid": "S3ListAllBuckets",
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "account-id"
        }
      }
    },
    {
      "Sid": "MemoryDbMultiRegionPauseReplication",
      "Effect": "Allow",
      "Action": [
        "memorydb:PauseMultiRegionClusterReplication",
        "memorydb:DescribeMultiRegionClusters"
      ],
      "Resource": "arn:aws:memorydb::account-id:multiregioncluster/*"
    },
    {
      "Sid": "TargetResolutionByTags",
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
    },
    {
      "Sid": "FISExperimentLogging",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs:DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": "*"
```

```
    },
    {
      "Sid": "FISExperimentLogGroupAccess",
      "Effect": "Allow",
      "Action": [
        "logs:DescribeResourcePolicies",
        "logs:PutResourcePolicy",
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

Multi-Region: recovery

The **Multi-Region: recovery** template runs AWS FIS actions that block dependency traffic in the impaired Region to exercise cross-Region failover. Attach the following permissions policy to the execution role.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Sid": "SsmSendCommandOnDocuments",
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ssm:*::document/AWSFIS-Run-Network-Packet-Loss-Sources",
        "arn:aws:ssm:*::document/AWSFIS-Run-Network-Packet-Loss-ECS"
      ]
    },
    {
      "Sid": "SsmSendCommandOnInstances",
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ec2:*:account-id:instance/*",
        "arn:aws:ssm:*:account-id:managed-instance/*",
        "arn:aws:ecs:*:account-id:task/*/*"
      ]
```

```
    },
    {
      "Sid": "SsmListAndCancelCommands",
      "Effect": "Allow",
      "Action": [
        "ssm:ListCommands",
        "ssm:CancelCommand"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Ec2DescribeForTargetResolution",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeSubnets"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EcsDescribeForTargetResolution",
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeTasks",
        "ecs:DescribeContainerInstances",
        "ecs:ListTasks"
      ],
      "Resource": [
        "arn:aws:ecs:*:account-id:task/*/*",
        "arn:aws:ecs:*:account-id:container-instance/*/*",
        "arn:aws:ecs:*:account-id:cluster/*"
      ]
    },
    {
      "Sid": "EksDescribeCluster",
      "Effect": "Allow",
      "Action": "eks:DescribeCluster",
      "Resource": "arn:aws:eks:*:account-id:cluster/*"
    },
    {
      "Sid": "TargetResolutionByTags",
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
```

```
    },
    {
      "Sid": "FISExperimentLogging",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs:DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": "*"
    },
    {
      "Sid": "FISExperimentLogGroupAccess",
      "Effect": "Allow",
      "Action": [
        "logs:DescribeResourcePolicies",
        "logs:PutResourcePolicy",
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

Create the role and attach it to your test

Save the trust policy as `trust-policy.json` and the permissions policy as `permissions-policy.json`, then create the role with the AWS CLI.

```
aws iam create-role \
  --role-name my-resilience-testing-role \
  --assume-role-policy-document file://trust-policy.json

aws iam put-role-policy \
  --role-name my-resilience-testing-role \
  --policy-name resilience-testing \
  --policy-document file://permissions-policy.json
```

Set the role on your test so that Resilience Hub uses it for test runs.

```
aws resiliencehubv2 update-test \
  --test-id test-id \
  --role-name my-resilience-testing-role
```

Multi-account execution roles

For a multi-account test, AWS FIS uses a role chain. AWS FIS assumes the *orchestrator role* in the account that runs the experiment. The orchestrator role then assumes a *target role* in each account that contains targeted resources. Create the orchestrator role once, and create a target role in every target account.

Orchestrator account role

Create the orchestrator role in the account that runs the experiment. Its trust policy allows AWS FIS to assume it, with the same confused-deputy protection as the single-account trust policy.

Trust policy

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Sid": "FISTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "fis.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "orchestrator-account-id"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:fis:*:orchestrator-account-id:experiment/*"
        }
      }
    },
    {
      "Sid": "SelfAssume",
      "Effect": "Allow",
      "Principal": {
```

```
        "AWS": "arn:aws:iam::orchestrator-account-id:role/orchestrator-role-name"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Permissions policy

The orchestrator permissions policy is the same for every test template. It allows the role to assume the target-account roles, resolve targets, and manage experiment logging and lifecycle. List the ARN of each target-account role in the `AssumeTargetAccountRole` statement.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Sid": "AssumeTargetAccountRole",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": [
        "arn:aws:iam::target-account-id:role/target-role-name"
      ]
    },
    {
      "Sid": "FISExperimentLogging",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs:DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": "*"
    },
    {
      "Sid": "FISExperimentLogGroupAccess",
      "Effect": "Allow",
      "Action": [
        "logs:DescribeResourcePolicies",
```

```
        "logs:PutResourcePolicy",
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    },
    {
      "Sid": "FISExperimentLifecycle",
      "Effect": "Allow",
      "Action": [
        "fis:GetExperiment",
        "fis:StopExperiment",
        "fis:ListExperimentResolvedTargets"
      ],
      "Resource": "arn:aws:fis:*:orchestrator-account-id:experiment/*"
    },
    {
      "Sid": "OrchestratorTagGetResources",
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
    },
    {
      "Sid": "OrchestratorDescribeForTargetResolution",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeTransitGateways",
        "ec2:DescribeTransitGatewayAttachments",
        "ec2:DescribeTransitGatewayPeeringAttachments",
        "ec2:DescribeRouteTables",
        "ec2:DescribeManagedPrefixLists",
        "ec2:GetManagedPrefixListEntries",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVolumes",
        "rds:DescribeDBClusters",
        "elasticache:DescribeReplicationGroups",
        "elasticache:DescribeCacheClusters",
        "ecs:DescribeTasks",
        "ecs:ListTasks",
        "ecs:DescribeServices",
        "ecs:DescribeContainerInstances",
```

```
        "ecs:DescribeClusters",
        "eks:DescribeCluster",
        "eks:ListClusters",
        "dynamodb:DescribeGlobalTable",
        "dynamodb:DescribeTable",
        "dynamodb:ListGlobalTables",
        "memorydb:DescribeMultiRegionClusters",
        "memorydb:DescribeClusters",
        "autoscaling:DescribeAutoScalingGroups",
        "elasticloadbalancing:DescribeLoadBalancers",
        "arc-zonal-shift:ListManagedResources",
        "arc-zonal-shift:GetManagedResource",
        "iam:GetRole",
        "iam:ListRoles",
        "s3:GetBucketTagging",
        "s3:ListAllMyBuckets",
        "s3:GetReplicationConfiguration",
        "ssm:DescribeInstanceInformation"
      ],
      "Resource": "*"
    }
  ]
}
```

Target account role

Create a target role in each account that contains targeted resources. Its trust policy allows the orchestrator role to assume it. The `sts:ExternalId` condition restricts the assume operation to AWS FIS experiments owned by the orchestrator account. The `aws:PrincipalArn` condition pins the trust to the orchestrator role.

Trust policy

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Sid": "OrchestratorRoleAssumeRole",
      "Effect": "Allow",
      "Principal": {
        "AWS": "orchestrator-account-id"
```

```
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringLike": {
          "sts:ExternalId": "arn:aws:fis:*:orchestrator-account-id:experiment/*"
        },
        "ArnEquals": {
          "aws:PrincipalArn": "arn:aws:iam::orchestrator-account-id:role/orchestrator-
role-name"
        }
      }
    }
  ]
}
```

Permissions policy

Attach the permissions policy for the test template that your test uses. These policies grant the fault-injection permissions for the targeted resources. Experiment logging and lifecycle permissions are granted by the orchestrator role, not the target role.

Topics

- Availability Zone: recovery
- Dependency validation
- Multi-Region: isolation
- Multi-Region: recovery

Availability Zone: recovery

Attach the following permissions policy to the target role for the **Availability Zone: recovery** template.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Sid": "EC2StopAndStartInstances",
      "Effect": "Allow",
      "Action": [
```

```
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": "arn:aws:ec2:*:target-account-id:instance/*"
    },
    {
      "Sid": "EC2DescribeInstances",
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    },
    {
      "Sid": "EC2EncryptedVolumesKmsGrant",
      "Effect": "Allow",
      "Action": "kms:CreateGrant",
      "Resource": "arn:aws:kms:*:target-account-id:key/*",
      "Condition": {
        "StringLike": {
          "kms:ViaService": "ec2.*.amazonaws.com"
        },
        "Bool": {
          "kms:GrantIsForAWSResource": "true"
        },
        "ForAllValues:StringEquals": {
          "kms:GrantOperations": ["Decrypt", "CreateGrant"]
        },
        "StringEquals": {
          "kms:GrantConstraintType": "EncryptionContextSubset"
        },
        "ForAnyValue:StringEquals": {
          "kms:EncryptionContextKeys": "aws:ebs:id"
        }
      }
    },
    {
      "Sid": "EC2InjectApiError",
      "Effect": "Allow",
      "Action": "ec2:InjectApiError",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "ec2:FisActionId": [
            "aws:ec2:api-insufficient-instance-capacity-error",
            "aws:ec2:asg-insufficient-instance-capacity-error"
```

```
          ]
        }
      }
    },
    {
      "Sid": "AutoScalingDescribe",
      "Effect": "Allow",
      "Action": [
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeTags"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ElastiCacheInterruptAzPower",
      "Effect": "Allow",
      "Action": [
        "elasticache:InterruptClusterAzPower",
        "elasticache:DescribeReplicationGroups"
      ],
      "Resource": "arn:aws:elasticache:*:target-account-id:replicationgroup:*"
    },
    {
      "Sid": "EBSPauseVolumeIO",
      "Effect": "Allow",
      "Action": "ec2:PauseVolumeIO",
      "Resource": "arn:aws:ec2:*:target-account-id:volume/*"
    },
    {
      "Sid": "EBSDescribeVolumes",
      "Effect": "Allow",
      "Action": "ec2:DescribeVolumes",
      "Resource": "*"
    },
    {
      "Sid": "NetworkTagManagedNacl",
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:*:target-account-id:network-acl/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkAcl",
          "aws:RequestTag/managedByFIS": "true"
        }
```

```
      }
    },
    {
      "Sid": "NetworkCreateManagedNacl",
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkAcl",
      "Resource": "arn:aws:ec2:*:target-account-id:network-acl/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "NetworkCreateNaclOnVpc",
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkAcl",
      "Resource": "arn:aws:ec2:*:target-account-id:vpc/*"
    },
    {
      "Sid": "NetworkModifyManagedNacl",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkAclEntry",
        "ec2:DeleteNetworkAcl"
      ],
      "Resource": "arn:aws:ec2:*:target-account-id:network-acl/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "NetworkDescribe",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeNetworkAcls",
        "ec2:DescribeManagedPrefixLists"
      ],
      "Resource": "*"
    },
```

```
    {
      "Sid": "NetworkReplaceNaclAssociation",
      "Effect": "Allow",
      "Action": "ec2:ReplaceNetworkAclAssociation",
      "Resource": [
        "arn:aws:ec2:*:target-account-id:subnet/*",
        "arn:aws:ec2:*:target-account-id:network-acl/*"
      ]
    },
    {
      "Sid": "NetworkPrefixListEntries",
      "Effect": "Allow",
      "Action": "ec2:GetManagedPrefixListEntries",
      "Resource": "arn:aws:ec2:*:target-account-id:prefix-list/*"
    },
    {
      "Sid": "RDSFailoverCluster",
      "Effect": "Allow",
      "Action": [
        "rds:FailoverDBCluster",
        "rds:DescribeDBClusters"
      ],
      "Resource": "arn:aws:rds:*:target-account-id:cluster:*"
    },
    {
      "Sid": "RDSDescribeForTargetResolution",
      "Effect": "Allow",
      "Action": "rds:DescribeDBInstances",
      "Resource": "arn:aws:rds:*:target-account-id:db:*"
    },
    {
      "Sid": "ARCZonalShiftManagedElb",
      "Effect": "Allow",
      "Action": [
        "arc-zonal-shift:StartZonalShift",
        "arc-zonal-shift:GetManagedResource",
        "arc-zonal-shift:UpdateZonalShift",
        "arc-zonal-shift:CancelZonalShift"
      ],
      "Resource": [
        "arn:aws:elasticloadbalancing:*:target-account-id:loadbalancer/app/*",
        "arn:aws:elasticloadbalancing:*:target-account-id:loadbalancer/net/*"
      ]
    },
```

```
    {
      "Sid": "ARCZonalShiftManagedAsgEks",
      "Effect": "Allow",
      "Action": [
        "arc-zonal-shift:StartZonalShift",
        "arc-zonal-shift:GetManagedResource",
        "arc-zonal-shift:UpdateZonalShift",
        "arc-zonal-shift:CancelZonalShift"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "arc-zonal-shift:ResourceIdentifier": [
            "arn:aws:autoscaling:*:target-account-id:autoScalingGroup:*",
            "arn:aws:eks:*:target-account-id:cluster/*"
          ]
        }
      }
    },
    {
      "Sid": "ARCZonalShiftList",
      "Effect": "Allow",
      "Action": "arc-zonal-shift:ListManagedResources",
      "Resource": "*"
    },
    {
      "Sid": "TargetResolutionByTags",
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
    }
  ]
}
```

Dependency validation

Attach the following permissions policy to the target role for the **Dependency validation** template.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
```

```
      "Sid": "SsmSendCommandOnDocuments",
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ssm:*::document/AWSFIS-Run-Network-Packet-Loss-Sources",
        "arn:aws:ssm:*::document/AWSFIS-Run-Network-Packet-Loss-ECS"
      ]
    },
    {
      "Sid": "SsmSendCommandOnInstances",
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ec2:*:target-account-id:instance/*",
        "arn:aws:ssm:*:target-account-id:managed-instance/*",
        "arn:aws:ecs:*:target-account-id:task/*/*"
      ]
    },
    {
      "Sid": "SsmListAndCancelCommands",
      "Effect": "Allow",
      "Action": [
        "ssm:ListCommands",
        "ssm:CancelCommand"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Ec2DescribeForTargetResolution",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeSubnets"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EcsDescribeForTargetResolution",
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeTasks",
        "ecs:DescribeContainerInstances",
        "ecs:ListTasks"
      ],
```

```
      "Resource": [
        "arn:aws:ecs:*:target-account-id:task/*/*",
        "arn:aws:ecs:*:target-account-id:container-instance/*/*",
        "arn:aws:ecs:*:target-account-id:cluster/*"
      ]
    },
    {
      "Sid": "EksDescribeCluster",
      "Effect": "Allow",
      "Action": "eks:DescribeCluster",
      "Resource": "arn:aws:eks:*:target-account-id:cluster/*"
    },
    {
      "Sid": "TargetResolutionByTags",
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
    }
  ]
}
```

Multi-Region: isolation

Attach the following permissions policy to the target role for the **Multi-Region: isolation** template.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Sid": "RouteTableDisruptCrossRegion",
      "Effect": "Allow",
      "Action": [
        "ec2:AssociateRouteTable",
        "ec2:DisassociateRouteTable",
        "ec2:ReplaceRouteTableAssociation",
        "ec2:CreateRoute"
      ],
      "Resource": [
        "arn:aws:ec2:*:target-account-id:route-table/*",
        "arn:aws:ec2:*:target-account-id:subnet/*"
      ]
    },
```

```
    {
      "Sid": "RouteTableManagedCreate",
      "Effect": "Allow",
      "Action": "ec2:CreateRouteTable",
      "Resource": "arn:aws:ec2:*:target-account-id:route-table/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "RouteTableManagedCreateOnVpc",
      "Effect": "Allow",
      "Action": "ec2:CreateRouteTable",
      "Resource": "arn:aws:ec2:*:target-account-id:vpc/*"
    },
    {
      "Sid": "RouteTableManagedDelete",
      "Effect": "Allow",
      "Action": "ec2:DeleteRouteTable",
      "Resource": [
        "arn:aws:ec2:*:target-account-id:route-table/*",
        "arn:aws:ec2:*:target-account-id:vpc/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "PrefixListManagedCreate",
      "Effect": "Allow",
      "Action": "ec2:CreateManagedPrefixList",
      "Resource": "arn:aws:ec2:*:target-account-id:prefix-list/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "PrefixListManagedModifyDelete",
```

```
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteManagedPrefixList",
        "ec2:ModifyManagedPrefixList"
      ],
      "Resource": "arn:aws:ec2:*:target-account-id:prefix-list/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "NetworkInterfaceManagedCreate",
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkInterface",
      "Resource": "arn:aws:ec2:*:target-account-id:network-interface/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "NetworkInterfaceManagedCreateOnSubnetAndSg",
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkInterface",
      "Resource": [
        "arn:aws:ec2:*:target-account-id:subnet/*",
        "arn:aws:ec2:*:target-account-id:security-group/*"
      ]
    },
    {
      "Sid": "NetworkInterfaceManagedDelete",
      "Effect": "Allow",
      "Action": "ec2:DeleteNetworkInterface",
      "Resource": "arn:aws:ec2:*:target-account-id:network-interface/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/managedByFIS": "true"
        }
      }
    },
    {
```

```
      "Sid": "FISCreateTags",
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": [
        "arn:aws:ec2:*:target-account-id:route-table/*",
        "arn:aws:ec2:*:target-account-id:prefix-list/*",
        "arn:aws:ec2:*:target-account-id:network-interface/*",
        "arn:aws:ec2:*:target-account-id:security-group/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Sid": "Ec2NetworkDescribeAndRead",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeRouteTables",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcPeeringConnections",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeManagedPrefixLists",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeSecurityGroups",
        "ec2:GetManagedPrefixListEntries",
        "ec2:DescribeTransitGateways",
        "ec2:DescribeTransitGatewayAttachments",
        "ec2:DescribeTransitGatewayPeeringAttachments",
        "ec2:DescribeInstances"
      ],
      "Resource": "*"
    },
    {
      "Sid": "TgwDisruptCrossRegion",
      "Effect": "Allow",
      "Action": [
        "ec2:AssociateTransitGatewayRouteTable",
        "ec2:DisassociateTransitGatewayRouteTable"
      ],
      "Resource": [
        "arn:aws:ec2:*:target-account-id:transit-gateway-route-table/*",
```

```
        "arn:aws:ec2:*:target-account-id:transit-gateway-attachment/*"
      ]
    },
    {
      "Sid": "VpcEndpointDisrupt",
      "Effect": "Allow",
      "Action": "ec2:ModifyVpcEndpoint",
      "Resource": [
        "arn:aws:ec2:*:target-account-id:vpc-endpoint/*",
        "arn:aws:ec2:*:target-account-id:security-group/*"
      ]
    },
    {
      "Sid": "VpcEndpointSecurityGroupManaged",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateSecurityGroup",
        "ec2:DeleteSecurityGroup",
        "ec2:RevokeSecurityGroupEgress"
      ],
      "Resource": [
        "arn:aws:ec2:*:target-account-id:security-group/*",
        "arn:aws:ec2:*:target-account-id:vpc/*"
      ]
    },
    {
      "Sid": "SsmSendCommandOnDocuments",
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ssm:*::document/AWSFIS-Run-Network-Packet-Loss-Sources",
        "arn:aws:ssm:*::document/AWSFIS-Run-Network-Packet-Loss-ECS"
      ]
    },
    {
      "Sid": "SsmSendCommandOnInstances",
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ec2:*:target-account-id:instance/*",
        "arn:aws:ssm:*:target-account-id:managed-instance/*",
        "arn:aws:ecs:*:target-account-id:task/*/*"
      ]
    },
```

```
    {
      "Sid": "SsmListAndCancelCommands",
      "Effect": "Allow",
      "Action": [
        "ssm:ListCommands",
        "ssm:CancelCommand"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EcsDescribeForTargetResolution",
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeTasks",
        "ecs:DescribeContainerInstances",
        "ecs:ListTasks"
      ],
      "Resource": [
        "arn:aws:ecs:*:target-account-id:task/*/*",
        "arn:aws:ecs:*:target-account-id:container-instance/*/*",
        "arn:aws:ecs:*:target-account-id:cluster/*"
      ]
    },
    {
      "Sid": "EksDescribeCluster",
      "Effect": "Allow",
      "Action": "eks:DescribeCluster",
      "Resource": "arn:aws:eks:*:target-account-id:cluster/*"
    },
    {
      "Sid": "DynamoDbGlobalTablePauseReplication",
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutResourcePolicy",
        "dynamodb:GetResourcePolicy",
        "dynamodb:DeleteResourcePolicy",
        "dynamodb:DescribeTable",
        "dynamodb:InjectError"
      ],
      "Resource": "arn:aws:dynamodb:*:target-account-id:table/*"
    },
    {
      "Sid": "S3PauseReplicationConfiguration",
      "Effect": "Allow",
```

```
      "Action": [
        "s3:PutReplicationConfiguration",
        "s3:GetReplicationConfiguration"
      ],
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "BoolIfExists": {
          "s3:IsReplicationPauseRequest": "true"
        }
      }
    },
    {
      "Sid": "S3PauseReplication",
      "Effect": "Allow",
      "Action": "s3:PauseReplication",
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Sid": "S3ListAllBuckets",
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    },
    {
      "Sid": "MemoryDbMultiRegionPauseReplication",
      "Effect": "Allow",
      "Action": [
        "memorydb:PauseMultiRegionClusterReplication",
        "memorydb:DescribeMultiRegionClusters"
      ],
      "Resource": "arn:aws:memorydb::target-account-id:multiregioncluster/*"
    },
    {
      "Sid": "TargetResolutionByTags",
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
    }
  ]
}
```

Multi-Region: recovery

Attach the following permissions policy to the target role for the **Multi-Region: recovery** template.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Sid": "SsmSendCommandOnDocuments",
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ssm:*::document/AWSFIS-Run-Network-Packet-Loss-Sources",
        "arn:aws:ssm:*::document/AWSFIS-Run-Network-Packet-Loss-ECS"
      ]
    },
    {
      "Sid": "SsmSendCommandOnInstances",
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ec2:*:target-account-id:instance/*",
        "arn:aws:ssm:*:target-account-id:managed-instance/*",
        "arn:aws:ecs:*:target-account-id:task/*/*"
      ]
    },
    {
      "Sid": "SsmListAndCancelCommands",
      "Effect": "Allow",
      "Action": [
        "ssm:ListCommands",
        "ssm:CancelCommand"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Ec2DescribeForTargetResolution",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeSubnets"
      ],
      "Resource": "*"
```

```
    },
    {
      "Sid": "EcsDescribeForTargetResolution",
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeTasks",
        "ecs:DescribeContainerInstances",
        "ecs:ListTasks"
      ],
      "Resource": [
        "arn:aws:ecs:*:target-account-id:task/*/*",
        "arn:aws:ecs:*:target-account-id:container-instance/*/*",
        "arn:aws:ecs:*:target-account-id:cluster/*"
      ]
    },
    {
      "Sid": "EksDescribeCluster",
      "Effect": "Allow",
      "Action": "eks:DescribeCluster",
      "Resource": "arn:aws:eks:*:target-account-id:cluster/*"
    },
    {
      "Sid": "TargetResolutionByTags",
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
    }
  ]
}
```

Create the roles and attach them to your test

In the orchestrator account, create the orchestrator role.

```
aws iam create-role \
  --role-name my-orchestrator-role \
  --assume-role-policy-document file://orchestrator-trust-policy.json

aws iam put-role-policy \
  --role-name my-orchestrator-role \
  --policy-name resilience-testing-orchestrator \
  --policy-document file://orchestrator-permissions-policy.json
```

In each target account, create the target role with the permissions policy for your test template.

```
aws iam create-role \
  --role-name my-target-role \
  --assume-role-policy-document file://target-trust-policy.json

aws iam put-role-policy \
  --role-name my-target-role \
  --policy-name resilience-testing-target \
  --policy-document file://target-permissions-policy.json
```

Set the orchestrator role as the execution role on your test.

```
aws resiliencehubv2 update-test \
  --test-id test-id \
  --role-name my-orchestrator-role
```

Dashboard and reporting

The Next generation Resilience Hub dashboard provides a centralized view of your organization's resilience posture, enabling SREs to monitor compliance, track trends, and generate reports across all services.

Topics

- Central SRE dashboard overview
- Service-level view
- Filtering by policy, account, Region, and system
- Generating compliance and resilience posture reports

Central SRE dashboard overview

The dashboard is your starting point for understanding resilience across your services. It provides:

- **Resilience posture summary** – Overall compliance status across all services
- **Assessment activity** – Recent and upcoming assessments
- **Findings overview** – Open failure mode findings by severity
- **Dependencies** – Newly discovered or unclassified dependencies

- **Test results** – Pass/fail status of resilience tests across your services

Service-level view

Drill into any service to see detailed information about its current state. The following sections are available in the service-level view.

Section	What it shows
Configuration	Input sources, permission model, associated system and user journeys
Topology	Visual map of resources and connections
Failure modes	Failure mode findings from the latest assessment with severity and status
Dependencies	Discovered dependencies with criticality and allowed status
History	Assessment history and resilience score trend

Filtering by policy, account, Region, and system

You can filter dashboard views using the following filters.

Filter	Description
Policy	Show only services using a specific resilience policy
Account	Focus on services in a specific AWS account
Region	Filter by AWS Region
System	Show services within a specific system
Severity	Filter failure mode findings by severity level
Compliance status	Compliant, non-compliant, or not assessed

For AWS Organizations users, the following additional filters are available:

- **Organizational Unit** – Filter by OU
- **Member account** – Focus on a specific member account

Generating compliance and resilience posture reports

Prerequisites

Before you can generate a report, your service must have:

- A **report output configuration** specifying the Amazon S3 bucket where reports are delivered.
- An **invoker role** that trusts the the next generation of Resilience Hub service and has permission to write to the configured Amazon S3 bucket.
- At least one **completed assessment** with a status of SUCCESS.

You can configure report outputs when creating or updating a service:

```
aws resiliencehubv2 update-service \
  --service-arn "arn:aws:resiliencehub:us-east-1:123456789012:service/my-
service:abc123" \
  --report-configuration '{"reportOutputs": [{"s3": {"bucketPath": "my-report-bucket",
 "bucketOwner": "123456789012"}}]}'
```

The invoker role must have a permissions policy that grants `s3:PutObject` on the target bucket. The following example shows the minimum required policy.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::my-report-bucket/*"
    }
  ]
}
```

> **Note**
>
> If your bucket uses a prefix in the `bucketPath` (for example, `my-report-bucket/reports`), scope the resource accordingly (for example, `arn:aws:s3:::my-report-bucket/reports/*`).

> **Note**
>
> If your Amazon S3 bucket is configured with SSE-KMS encryption, the invoker role also needs `kms:GenerateDataKey` and `kms:Encrypt` permissions on the bucket's KMS key.

Generating a failure mode assessment report

To generate a report after running an assessment:

1. Navigate to your service.
2. Choose the **Assessment** tab.
3. Choose **Generate report**.

Test run reports

Test reports are auto-generated after each test run completes if a report destination is configured. No additional steps are needed. For details on what's included, see Test runs and report.

Viewing reports

In the left-hand navigation, choose **Reports**. The reports page shows all generated reports (failure mode assessments and test runs) that you have access to.

Working with AWS DevOps Agent in the next generation of Resilience Hub

The next generation of Resilience Hub integrates with AWS DevOps Agent to add agentic investigation to your failure mode assessment findings and resilience test runs. When you connect

a service to an *Agent Space* in AWS DevOps Agent, Resilience Hub can launch investigations on your behalf:

- From any failure mode assessment finding, you can chat with AWS DevOps Agent to investigate and ask for remediation steps interactively.
- For a resilience test, a DevOps Agent investigation starts automatically with the test run, and a summary of its analysis becomes available shortly after the test completes – shown on the test run and included in the test report.

Resilience Hub calls AWS DevOps Agent on your behalf by assuming the same *invoker role* that you have already set up at service level. AWS DevOps Agent investigation charges apply.

For more information about AWS DevOps Agent, see the AWS DevOps Agent User Guide.

Topics

- Prerequisites
- Add DevOps Agent permissions to the invoker role
- Connect a service to the Agent Space
- Investigate findings and test runs

Prerequisites

Before you connect a service, make sure that you have the following:

- A DevOps Agent *Agent Space*. You create and configure the Agent Space in the DevOps Agent console. Note its Amazon Resource Name (ARN), which has the form `arn:aws:aidevops:`*`region`*`:`*`account-id`*`:agentspace/`*`agent-space-id`*.
- An existing Next generation Resilience Hub service, with resource discovery already configured. The cross-account invoker role that discovery uses must exist. You add DevOps Agent permissions to this same role in the section called “Add DevOps Agent permissions to the invoker role”.
- An Agent Space in a DevOps Agent supported AWS Region. DevOps Agent can investigate resources in any Region regardless of where the Agent Space is created.

Add DevOps Agent permissions to the invoker role

Resilience Hub connects to your Agent Space by assuming your existing invoker role – the same cross-account role that resource discovery uses. You do *not* create a new role or change its trust policy. You only add the following DevOps Agent (`aidevops`) permissions to it.

If the invoker role is missing these permissions, the connection fails with an access denied error, and the service's DevOps Agent status becomes `FAILED` with a reason that tells you to update the role. This is the most common cause of a failed connection.

Permissions policy

Attach the following least-privilege policy to the invoker role. These eight actions are the complete set that Resilience Hub calls on your behalf – validating connectivity and starting, polling, and canceling investigations. None of them can mutate your Agent Space, or your resources.

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Sid": "ResilienceHubV2DevOpsAgentInvestigation",
      "Effect": "Allow",
      "Action": [
        "aidevops:CreateBacklogTask",
        "aidevops:GetBacklogTask",
        "aidevops:ListExecutions",
        "aidevops:ListJournalRecords",
        "aidevops:UpdateBacklogTask"
      ],
      "Resource": "arn:aws:aidevops:*:*:agentspace/*"
    },
    {
      "Sid": "ResilienceHubV2DevOpsAgentValidation",
      "Effect": "Allow",
      "Action": [
        "aidevops:GetAgentSpace",
        "aidevops:GetOperatorApp",
        "aidevops:ListAssociations"
      ],
      "Resource": "arn:aws:aidevops:*:*:agentspace/*"
    }
  ]
```

```
}
```

The policy contains two statements:

- `ResilienceHubV2DevOpsAgentInvestigation` – Lets Resilience Hub start, poll, update, and cancel the backlog tasks and read the journal records that make up an investigation.
- `ResilienceHubV2DevOpsAgentValidation` – Grants the read-only access that Resilience Hub uses to validate the Agent Space connection.

> **Important**
>
> Do not attach the broad AWS managed policies `AIDevOpsAgentFullAccess` or `AIDevOpsOperatorAppAccessPolicy`. They grant mutating Agent Space actions that Resilience Hub must never call. The inline policy above is the only grant that the integration needs.

Add the policy with the AWS CLI

Save the permissions policy as `resiliencehub-devops-agent-policy.json`, then add it to the invoker role as an inline policy.

```
aws iam put-role-policy \
  --role-name my-invoker-role \
  --policy-name ResilienceHubDevOpsAgentAccess \
  --policy-document file://resiliencehub-devops-agent-policy.json
```

To confirm that the policy was added, run the following command.

```
aws iam get-role-policy \
  --role-name my-invoker-role \
  --policy-name ResilienceHubDevOpsAgentAccess
```

Connect a service to the Agent Space

After your invoker role has the required DevOps Agent permissions, you can connect a service to your Agent Space. Connect a service by providing the Agent Space ARN in the service's DevOps Agent configuration. You can do this when you create a service or when you update an existing

one. If there are existing failure mode assessment findings, Resilience Hub runs investigations for each finding after a connection is first established.

The following command connects an existing service to an Agent Space.

```
aws resiliencehubv2 update-service \
  --service-arn service-arn \
  --dev-ops-agent-configuration '{"agentSpaceArn": "agent-space-arn"}'
```

To check the connection status, call `GetService` and inspect the `devOpsAgentConfiguration` field.

```
aws resiliencehubv2 get-service \
  --service-arn service-arn \
  --query 'service.devOpsAgentConfiguration.status'
```

To disconnect a service later, update it with an empty DevOps Agent configuration (`'{}'`). This clears the connection without affecting the Agent Space itself.

Investigate findings and test runs

After a service is connected, you can launch AWS DevOps Agent directly from the the next generation of Resilience Hub console. When you view a specific failure mode assessment finding, you see an option to chat with DevOps Agent about that finding. For resilience tests, a DevOps Agent investigation starts automatically with the test run, and a summary becomes available shortly after the test completes.

- **Failure mode assessment findings** – Investigate a finding with DevOps Agent by choosing **Chat with DevOps Agent**, which is available on every failure mode assessment finding. The console opens an interactive DevOps Agent session seeded with that finding. If you use the API, each finding returns an `investigationDeepLink` to the same session.
- **Resilience test runs** – When a test run finishes, open the test run in the console to see the DevOps Agent investigation, including its verdict, confidence, and analysis summary, and choose the link to open the full session. If you use the API, get the test run and inspect the `devOpsAgentInvestigation` block. The DevOps Agent verdict is advisory; your test run's own result remains authoritative.
- **Test reports** – When a test report is generated for a connected service, it includes a DevOps Agent investigation section with the summary. The test report generation completes when DevOps Agent completes the investigation.

AWS Organizations integration

Next generation Resilience Hub integrates with AWS Organizations to enable organization-wide resilience management from a single delegated administrator (DA) account. This eliminates the need to log into individual accounts to assess resilience posture across your enterprise. You get organization-wide visibility into resilience posture and aggregated dashboards without logging into individual accounts.

Metric	Value
Accounts per organization	50,000+
Services across organization	10,000
Dashboard query latency	Independent of organization size
Data aggregation latency	Less than 5 minutes

Topics

- Overview of multi-account governance in the next generation of Resilience Hub
- Setting up Organizations integration
- Service-Linked Roles
- Applying resilience policies across accounts
- Tracking changes to a shared policy
- Viewing organization-wide resilience posture
- Managing member account onboarding
- Required IAM permissions for delegated administrator setup

Overview of multi-account governance in the next generation of Resilience Hub

Next generation Resilience Hub enables centralized resilience governance across your AWS organization. With Organizations integration, you can:

- View resilience posture across all accounts from a single dashboard.

- Create and publish organization-wide resilience policies.
- Monitor compliance across hundreds of accounts and services.
- Filter by account, AWS Region, organizational unit (OU), and policy.

The following core concepts apply to the Organizations integration model:

Concept	Description
Delegated administrator	A member account designated to manage the next generation of Resilience Hub across the organization
Org-level policies	Resilience policies created by the DA, visible and assignable across all member accounts
Service-Linked Roles	Automatically created in member accounts for read-only cross-account access

In AWS Organizations, the delegated administrator:

- Has visibility into all systems and services across all member accounts.
- Creates and publishes organization-wide resilience policies by associating them with user journeys on shared systems.
- Views aggregated resilience posture dashboards.

Service-Linked Roles (SLRs) are automatically created in all member accounts when trusted access is enabled, providing the DA with read-only cross-account visibility without manual IAM setup.

Setting up Organizations integration

To use the next generation of Resilience Hub with AWS Organizations, you must designate a delegated administrator account that manages organization-wide policies and visibility.

Prerequisites

- AWS Organizations must be configured with all features enabled.
- You must have access to the management account to enable trusted access.

- Identify which member account will serve as the delegated administrator.
- The management account must create the Service-Linked Role (SLR) for the Organizations integration to function. Without the SLR, the delegated administrator cannot access member account data.

> **Important**
>
> The management account must create the Service-Linked Role (SLR) during setup. The Organizations integration does not function without the SLR, and the delegated administrator cannot access member account data until the SLR is created.

Quick setup steps

1. From the **management account**, open the the next generation of Resilience Hub console and choose **AWS Organizations settings**.
2. Select the checkboxes to enable trusted access and create the Service-Linked Role (SLR).
3. Choose **Create integration**.
4. (Optional) Register a delegated administrator by choosing **Register** on the same page.
5. From the **delegated administrator account**, select the home region in the the next generation of Resilience Hub console.
6. Individual service owners in member accounts create their own invoker roles for their services. For invoker role setup, see Setting up Next generation Resilience Hub.

After setup completes, Next generation Resilience Hub performs the following actions:

1. Service-Linked Roles are created in all member accounts. This may take additional time for large organizations due to API throttling.
2. Next generation Resilience Hub fetches the initial organization structure.
3. The DA account gains cross-account visibility through the SLRs.
4. Organization structure sync begins (event-driven plus a 12-hour reconciliation job).

Service-Linked Roles

Service-Linked Roles (`AWSServiceRoleForResilienceHub`) are IAM roles that are automatically created in every member account when you enable trusted access for `resiliencehub.amazonaws.com` from the management account. These roles provide the delegated administrator with read-only cross-account visibility into member account resources without requiring manual IAM configuration.

SLRs are created automatically when a new account joins the organization.

Individual service owners in member accounts still create their own invoker roles for running assessments on their services. The SLR provides cross-account visibility for the DA; it does not replace the invoker role used for discovery and assessment. For invoker role setup, see Setting up Next generation Resilience Hub.

Applying resilience policies across accounts

A management account or delegated administrator can apply resilience policies across member accounts by associating a policy with a system and then associating services in member accounts to that system. The system-level policy is applied to all associated services.

1. Create a resilience policy in the management account or DA account.
2. Create a system and associate the policy with it.
3. Associate services from member accounts to the system.

The system-level policy applies to all services associated with that system, regardless of which member account owns the service.

You can also share a policy directly with your organization instead of applying it through a system. A shared policy is visible in every member account, and service owners choose whether to apply it to their services. Share a policy directly when service teams own the decision to adopt it. Associate the policy with a system when you want it applied to every associated service, regardless of which account owns the service. For more information, see Sharing a policy across your organization.

Tracking changes to a shared policy

A shared policy keeps a timeline of the events that affected it. The account that owns the policy can see how member accounts used it and what changed. Use the `ListPolicyEvents` API operation to retrieve the timeline:

```
aws resiliencehubv2 list-policy-events \
  --policy-arn "arn:aws:resiliencehub:us-east-1:111122223333:policy/tier-1-dr:abc123"
```

The timeline records the following events.

Event	Description
Policy attached to service	The event records the service that started using the policy, and the AWS account that owns the service.
Policy detached from service	The event records the service that stopped using the policy, and the reason it stopped.
Policy sharing revoked	The event records that the policy owner turned off cross-acc ount sharing, and the number of services using the policy at that moment.
Policy deleted	The event records that the policy owner deleted the policy, and the number of services using it at that moment.

A service stops using a policy for one of three reasons. The service owner applied a different policy, the policy owner turned off cross-account sharing, or the policy owner deleted the policy.

You can filter the timeline by time range and by event type. Only the account that owns the policy can read its timeline. A member account that uses a shared policy cannot read it.

The last two events report a point-in-time count. Next generation Resilience Hub captures the number of services at the moment of the change, so the count does not update afterward.

Viewing organization-wide resilience posture

The delegated administrator sees aggregated views across the entire organization. The organization dashboard shows:

- Total services across all accounts with their compliance status.
- Findings summary by severity across the organization.

- Assessment activity and trends.
- Services without policies or recent assessments.

Managing member account onboarding

Member account onboarding is handled automatically through SLRs:

- **New accounts** – When a new account joins the organization, an SLR is automatically created in the new account, the account appears in the DA's organization view, and services created in the new account are visible to the DA.
- **Removed accounts** – When an account leaves the organization, data for that account is no longer visible to the DA, and services in the removed account continue to function independently.

Required IAM permissions for delegated administrator setup

The following IAM permissions are required for each role in the Organizations integration:

Management account

The management account needs permissions to:

- `organizations:EnableAWSServiceAccess`
- `organizations:RegisterDelegatedAdministrator`
- `iam:CreateServiceLinkedRole` (for the management account's own SLR)

Delegated administrator account

The DA account uses standard the next generation of Resilience Hub API permissions. Cross-account access is handled by SLRs – no additional IAM configuration is needed for viewing member account data.

Member accounts

Service owners in member accounts:

- Create their own invoker roles using the same process as the single-account setup. For details, see Setting up Next generation Resilience Hub.

- Can see and apply org-level policies published by the DA.
- The SLR handles DA cross-account visibility automatically – no additional IAM changes are required in member accounts.

The following table summarizes what the DA can and cannot do:

Action	Supported
View member account services, findings, and dependencies	Yes
Create org-level systems that reference member services	Yes
Associate member services to org-level systems	Yes
Create org-level policies	Yes
Delete member account services	No
Start assessments on member services	Yes
Modify member account resources	No

Destructive operations on member resources are not supported through DA cross-account access. The DA manages org-level systems and policies and views member data.

Migrating from AWS Resilience Hub

If you're an existing AWS Resilience Hub (v1) customer, this guide helps you understand what changes with the next generation of Resilience Hub and how to migrate your applications. The following table summarizes key changes between the two versions.

Area	AWS Resilience Hub v1	Next generation Resilience Hub
Core primitive	Application	System + Services
Assessment engine	Static rule-based checks	GenAI-powered failure mode analysis

Area	AWS Resilience Hub v1	Next generation Resilience Hub
Dependency visibility	None	Dependency discovery
Multi-account	Limited	Full AWS Organizations integration
Policies	Single RTO/RPO policy per application	Modular, composable policies (DR + Availability + Data recovery)
Testing	AWS FIS experiment templates (manual setup)	Recommended resilience tests (pre-configured, auto-targeted, pass/fail)
API version	/v1	/v2

Topics

- What changes with Next generation Resilience Hub
- Concept mapping: AWS Resilience Hub v1 to Next generation Resilience Hub
- Step-by-step migration guide
- Pricing transition
- Known limitations during migration

What changes with Next generation Resilience Hub

Next generation Resilience Hub introduces a new application modeling hierarchy, GenAI-powered assessments, and automatic dependency discovery. This section summarizes the key changes.

- **New hierarchy** – The v1 concept of an "application" maps to a "service" as the primary unit of assessment. Services are grouped within systems, and user journeys describe the paths through them.
- **GenAI-powered failure mode assessments** – Static rule-based checks are replaced by generative AI analysis that produces detailed failure mode findings with reasoning specific to your architecture.
- **Dependency discovery** – Next generation Resilience Hub automatically discovers dependencies between services and external resources, a capability not available in v1.

- **Modular resilience policies** – Policies are now composable. You can combine disaster recovery (DR), availability SLO, and data recovery conditions in a single policy, rather than a single RTO/RPO pair.
- **Full AWS Organizations integration** – Organization-wide governance from a single delegated administrator account replaces the limited multi-account support in v1.

Concept mapping: AWS Resilience Hub v1 to Next generation Resilience Hub

The following table maps concepts from AWS Resilience Hub v1 to their equivalents in the next generation of Resilience Hub.

AWS Resilience Hub v1 concept	Next generation Resilience Hub concept	Notes
Application	Service	Your v1 "application" becomes a the next generation of Resilience Hub "service" – the primary unit of assessment
Resilience checks	Failure mode assessment findings	Static checks replaced by GenAI-pow ered findings with reasoning
Application assessment	Service failure mode assessment	Same concept, now at service level with richer output
Assessment policy (RTO/RPO)	Resilience policy (modular)	Policies are now composable: DR + Availability SLO + Data recovery
Application (as grouping)	System + User journeys	The "grouping" aspect of applications maps to systems and user journeys
– (not available)	Dependency discovery	New capability in the next generation of Resilience Hub

AWS Resilience Hub v1 concept	Next generation Resilience Hub concept	Notes
– (not available)	Service functions	Technical workflows within a service; new in the next generation of Resilienc e Hub

Step-by-step migration guide

Follow these steps to migrate your existing AWS Resilience Hub v1 applications to the next generation of Resilience Hub.

Identify the application to migrate

Use the AWS Resilience Hub v1 API to find the application ARN you want to migrate.

```
# List your v1 applications
aws resiliencehub list-apps
```

Note the ARN of the application you want to migrate. You use this ARN in the next step.

Use the migration API

Next generation Resilience Hub provides migration APIs that handle the transition for you. The `import-app` API creates the service automatically – you don't need to manually create a system or service first.

```
# Import a v1 application to a next generation Resilience Hub service
aws resiliencehubv2 import-app \
  --v1-app-arn "arn:aws:resiliencehub:us-east-1:123456789012:app/my-app:..."

# Import a v1 policy to a next generation Resilience Hub policy
aws resiliencehubv2 import-policy \
  --v1-policy-arn "arn:aws:resiliencehub:us-east-1:123456789012:resiliency-policy/..."
```

The migration API performs the following actions:

- Creates a service from your v1 application (within a new or existing system)
- Preserves your input sources and resource configuration

- Converts your v1 policy to a resilience policy with modular conditions
- Maintains assessment history for continuity

Run your first failure mode assessment

After migration, run a failure mode assessment to see the enhanced output.

```
aws resiliencehubv2 start-failure-mode-assessment \
  --service-arn "arn:aws:resiliencehub:..."
```

The failure mode findings include detailed reasoning specific to your architecture, recommendations with cost and complexity guidance, and policy mapping showing which requirements are at risk.

Enable new capabilities

Take advantage of features not available in v1.

1. **Enable Dependency discovery** – Discover hidden dependencies between services and external resources.
2. **Set up Organizations** – Enable multi-account governance across your organization.
3. **Add performance conditions** – Define what "available" means beyond uptime using availability SLO conditions.

Pricing transition

The following table shows how pricing components map from AWS Resilience Hub v1 to the next generation of Resilience Hub.

Component	AWS Resilience Hub v1	Next generation Resilience Hub
Base fee	$15/application/month	$15/service/month
Assessments included	Unlimited	2 per month
Resource overage	None	$0.10/resource above 150 per assessment

Component	AWS Resilience Hub v1	Next generation Resilience Hub
Dependency discovery	Not available	$10/service/month (optional)

The $15 entry price is preserved. Most customers (services with 150 or fewer resources using 2 or fewer assessments per month) pay the same amount.

Known limitations during migration

The following limitations apply during the migration period from AWS Resilience Hub v1 to the next generation of Resilience Hub.

Limitation	Description	Workaround
Assessment history	v1 assessment results are not automatically migrated to the next generation of Resilience Hub format	v1 results remain accessible through v1 APIs during the transition period
Custom resilience checks	v1 custom checks are not migrated	Review failure mode findings – GenAI assessments typically cover the same concerns
AppRegistry input source	AppRegistry is not supported as an input source in the next generation of Resilience Hub	Use alternative input sources such as CloudFormation stacks or resource tags

Security in Next generation Resilience Hub

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud.

This topic describes security features and best practices for Next generation Resilience Hub, including IAM permissions, data encryption, network security, and audit logging.

Topics

- IAM roles and permissions reference
- Data encryption
- VPC endpoints
- CloudTrail integration
- Compliance considerations

IAM roles and permissions reference

IAM role for assessment and resilience testing

To run an assessment or a resilience test, the next generation of Resilience Hub needs to be able to assume an IAM role. Assessments require read-only permissions to discover and understand the configuration of your AWS resources. Resilience testing additionally requires permissions to start and manage AWS Fault Injection Service (AWS FIS) experiments on your behalf.

You can create an IAM role in the AWS IAM console. Choose **Custom trust policy** and use a trust policy like this:

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

For permissions, attach the `AWSResilienceHubV2AssessmentExecutionPolicy` managed policy. This policy grants read-only access to AWS services for resilience discovery, assessment, and management.

> **Note**
>
> The `AWSResilienceHubV2AssessmentExecutionPolicy` replaces the previous `AWSResilienceHubAsssessmentExecutionPolicy` for use with the next generation of Resilience Hub. For details about the permissions included in this policy, see AWSResilienceHubV2AssessmentExecutionPolicy.

If you use resilience testing, also attach the `AWSResilienceHubResilienceTestingPolicy` managed policy. This policy grants Resilience Hub the AWS FIS permissions needed to start and manage experiments on your behalf.

> **Note**
>
> If the managed policy is not available in your account, create an inline policy with the same permissions. For the policy contents, see AWSResilienceHubResilienceTestingPolicy.

IAM Service-Linked Role

Next generation Resilience Hub automatically creates a Service-Linked Role with the `AWSResilienceHubServiceRolePolicy` managed policy.

Terraform state file access permissions

If you are including Terraform state files into your Next generation Resilience Hub service, provide permissions to read the Terraform files from your Amazon S3 bucket with a policy like this:

```
{
  "Version": "2012-10-17"       ,
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::s3-bucket-name/path-to-state-file"
    },
```

```
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::s3-bucket-name"
    }
  ]
}
```

IAM execution role for resilience testing

Resilience tests run under an IAM role that AWS Fault Injection Service assumes to inject faults into your resources. The required permissions depend on the test template, and single-account and multi-account tests use different role structures. For the trust and permissions policies for each test, see IAM execution roles for resilience testing.

Amazon EKS Permissions

If you are including Amazon EKS clusters into your Next generation Resilience Hub service, follow the following 3-step process to provide Next generation Resilience Hub permissions to read configuration data for your Amazon EKS clusters using Kubernetes role-based access control (RBAC).

Step 1: Apply the following to your Amazon EKS cluster

This grants Next generation Resilience Hub read-only access to the Kubernetes resources it needs across all namespaces:

```
cat << EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: resilience-hub-eks-access-cluster-role
rules:
- apiGroups:
    - ""
  resources:
    - pods
    - replicationcontrollers
    - nodes
    - services
  verbs:
```

```
    - get
    - list
- apiGroups:
    - apps
  resources:
    - deployments
    - replicasets
  verbs:
    - get
    - list
- apiGroups:
    - networking.k8s.io
  resources:
    - ingresses
  verbs:
    - get
    - list
- apiGroups:
    - policy
  resources:
    - poddisruptionbudgets
  verbs:
    - get
    - list
- apiGroups:
    - autoscaling.k8s.io
  resources:
    - verticalpodautoscalers
  verbs:
    - get
    - list
- apiGroups:
    - autoscaling
  resources:
    - horizontalpodautoscalers
  verbs:
    - get
    - list
- apiGroups:
    - karpenter.sh
  resources:
    - provisioners
    - nodepools
  verbs:
```

```
    - get
    - list
- apiGroups:
    - karpenter.k8s.aws
  resources:
    - awsnodetemplates
    - ec2nodeclasses
  verbs:
    - get
    - list
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: resilience-hub-eks-access-cluster-role-binding
subjects:
  - kind: Group
    name: resilience-hub-eks-access-group
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: resilience-hub-eks-access-cluster-role
  apiGroup: rbac.authorization.k8s.io
---
EOF
```

Step 2: Map the IAM role to the Kubernetes group

Map the IAM role you created to the `resilience-hub-eks-access-group` Kubernetes group. You can use either Amazon EKS access entries (recommended) or the `aws-auth` ConfigMap.

Option A: Using EKS access entries (recommended)

EKS access entries are the preferred method for managing cluster authentication. Your cluster must use `API` or `API_AND_CONFIG_MAP` authentication mode.

```
aws eks create-access-entry \
  --cluster-name cluster-name \
  --principal-arn arn:aws:iam::ACCOUNT-ID:role/ResilienceHubRole \
  --type STANDARD \
  --kubernetes-groups '["resilience-hub-eks-access-group"]'
```

Option B: Using aws-auth ConfigMap

If your cluster uses `CONFIG_MAP` or `API_AND_CONFIG_MAP` authentication mode, you can edit the aws-auth ConfigMap instead:

Using eksctl:

```
eksctl create iamidentitymapping \
  --cluster cluster-name \
  --region region \
  --arn arn:aws:iam::ACCOUNT-ID:role/ResilienceHubRole \
  --group resilience-hub-eks-access-group \
  --username AwsResilienceHubAssessmentEKSAccessRole
```

Or manually edit the ConfigMap:

```
kubectl edit -n kube-system configmap/aws-auth
```

Add this under `mapRoles` in the data section:

```
- groups:
    - resilience-hub-eks-access-group
  rolearn: arn:aws:iam::ACCOUNT-ID:role/ResilienceHubRole
  username: AwsResilienceHubAssessmentEKSAccessRole
```

Step 3: Verify

Confirm the RBAC resources exist and the role mapping is in place:

```
kubectl get clusterrole resilience-hub-eks-access-cluster-role
kubectl describe clusterrolebinding resilience-hub-eks-access-cluster-role-binding
```

If using access entries (Option A):

```
aws eks describe-access-entry \
  --cluster-name cluster-name \
  --principal-arn arn:aws:iam::ACCOUNT-ID:role/ResilienceHubRole
```

If using aws-auth ConfigMap (Option B):

```
kubectl get configmap aws-auth -n kube-system -o yaml | grep -A 3 "ResilienceHubRole"
```

Data encryption

Next generation Resilience Hub encrypts your data to help protect it from unauthorized access.

Encryption at rest

All Next generation Resilience Hub data is encrypted at rest.

Data store	Encryption
DynamoDB tables	AWS-managed keys (default) or customer-managed AWS KMS keys
S3 objects (topology, assessment results)	SSE-S3 (default) or SSE-KMS with customer-managed keys

Encryption at rest with customer managed keys

Next generation Resilience Hub provides encryption by default to protect sensitive customer data at rest using AWS owned keys.

- Next generation Resilience Hub uses these keys by default to automatically encrypt sensitive data. You cannot view, manage, or use AWS owned keys, or audit their use. However, you do not have to take any action or change any programs to protect the keys that encrypt your

data. For more information, see https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#aws-owned-cmk in the *AWS Key Management Service Developer Guide*.

While you cannot disable this layer of encryption or select an alternate encryption type, you can add a second layer of encryption by specifying a customer managed key when you create a service resource:

- **Customer managed keys** Next generation Resilience Hub supports the use of a symmetric encryption customer managed key that you create, own, and manage. Because you have full control of this layer of encryption, you can perform such tasks as:
  - Establishing and maintaining key policies
  - Establishing and maintaining IAM policies and grants
  - Enabling and disabling key policies
  - Rotating key cryptographic material
  - Adding tags
  - Creating key aliases
  - Scheduling keys for deletion

  For more information, see Customer managed keys in the *AWS Key Management Service Developer Guide*.

The following table summarizes how Next generation Resilience Hub encrypts sensitive data.

Encryption of data types in Next generation Resilience Hub

Data type	AWS owned key encryption	Customer managed key encryption (optional)
`description` Descriptions for services, systems, and resiliency policies.	Enabled	Enabled
`finding`	Enabled	Enabled

Data type	AWS owned key encryption	Customer managed key encryption (optional)
Assessment finding names, descriptions, reasoning, and comments.		
`recommendation` Recommendation descriptions and suggested changes associated with findings.	Enabled	Enabled
`serviceFunction` Service function names and descriptions.	Enabled	Enabled
`assertion` Assertion text (AI-generated and user-crea ted).	Enabled	Enabled
`userJourney` User journey descriptions.	Enabled	Enabled
`event` Service event log descriptions.	Enabled	Enabled
`assessmentData` Intermediate data generated by agentic assessment workflows, including topology, resource configuration, and working data stored in Amazon S3.	Enabled	Enabled

Data type	AWS owned key encryption	Customer managed key encryption (optional)
Resource identifiers Resource names, ARNs, resource types, and regions. Resource names are used in identifiers and encryption context and must not contain sensitive data.	Enabled	Not supported

> **Note**
>
> Next generation Resilience Hub automatically enables encryption at rest using AWS owned keys at no charge. However, AWS KMS charges apply for using a customer managed key. For more information about pricing, see AWS Key Management Service pricing.

> **Important**
>
> Next generation Resilience Hub supports only symmetric encryption KMS keys. You cannot use any other type of KMS key to encrypt your Next generation Resilience Hub resources. For help determining whether a KMS key is a symmetric encryption key, see Identifying symmetric and asymmetric KMS keys in the *AWS Key Management Service Developer Guide*.

How Next generation Resilience Hub uses grants in AWS KMS

Next generation Resilience Hub requires a grant to use your customer managed key during asynchronous assessment workflows.

When you create a service with a customer managed key, Next generation Resilience Hub creates a grant on your behalf by sending a CreateGrant request to AWS KMS. The grant is constrained to the encryption context of your service and allows only the following operations:

- `Encrypt` – Encrypt sensitive fields such as findings, recommendations, and assertions generated during assessment workflows.

- `Decrypt` – Decrypt previously encrypted data during assessment processing.
- `GenerateDataKey` – Generate data keys for encrypting intermediate assessment data stored in Amazon S3.

The grant is retired when you delete the service. You can also revoke access to the grant, or remove the service's access to the customer managed key at any time. If you do, Next generation Resilience Hub cannot access any of the data encrypted by the customer managed key, which affects API operations and assessment workflows that depend on that data.

For synchronous API operations (such as creating or updating a service), Next generation Resilience Hub uses the caller's permissions on the KMS key directly, without requiring a grant.

Create a customer managed key

You can create a symmetric encryption customer managed key by using the AWS Management Console or the AWS KMS APIs.

To create a symmetric encryption customer managed key

Follow the steps for Creating symmetric encryption KMS keys in the *AWS Key Management Service Developer Guide*.

Specifying a customer managed key for Next generation Resilience Hub

You can specify a customer managed key when you create a service, system, or resiliency policy. When you provide a KMS key ID, Next generation Resilience Hub uses that key to encrypt all sensitive data associated with the resource.

You can specify the key using any of the following key identifiers:

- Key ID (for example, `1234abcd-12ab-34cd-56ef-1234567890ab`)
- Key ARN (for example, `arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`)
- Alias name (for example, `alias/my-key`)
- Alias ARN (for example, `arn:aws:kms:us-west-2:111122223333:alias/my-key`)

To specify a customer managed key, use the `kmsKeyId` parameter when calling the `CreateService`, `CreateSystem`, or `CreatePolicy` API operations.

Key policy

Key policies control access to your customer managed key. Every customer managed key must have exactly one key policy, which contains statements that determine who can use the key and how they can use it. When you create your customer managed key, you can specify a key policy. For more information, see Managing access to customer managed keys in the *AWS Key Management Service Developer Guide*.

The following key policy allows Next generation Resilience Hub to use your key. It scopes each permission to only the operations Next generation Resilience Hub requires, using encryption context conditions to ensure your key can only be used for your specific resources. Replace *CUSTOMER-ACCOUNT-ID*, *CUSTOMER-ROLE*, and *REGION* with your values.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowResilienceHubDescribeKey",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::CUSTOMER-ACCOUNT-ID:role/CUSTOMER-ROLE"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "resiliencehub.REGION.amazonaws.com"
                }
            }
        },
        {
            "Sid": "AllowResilienceHubEncryptDecryptForServices",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::CUSTOMER-ACCOUNT-ID:role/CUSTOMER-ROLE"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
```

```
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "resiliencehub.REGION.amazonaws.com"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:resiliencehub:service-arn":
"arn:aws:resiliencehub:*:CUSTOMER-ACCOUNT-ID:service/*"
                }
            }
        },
        {
            "Sid": "AllowResilienceHubEncryptDecryptForSystems",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::CUSTOMER-ACCOUNT-ID:role/CUSTOMER-ROLE"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "resiliencehub.REGION.amazonaws.com"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:resiliencehub:system-arn":
"arn:aws:resiliencehub:*:CUSTOMER-ACCOUNT-ID:system/*"
                }
            }
        },
        {
            "Sid": "AllowResilienceHubEncryptDecryptForPolicies",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::CUSTOMER-ACCOUNT-ID:role/CUSTOMER-ROLE"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
```

```
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "resiliencehub.REGION.amazonaws.com"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:resiliencehub:policy-arn":
 "arn:aws:resiliencehub:*:CUSTOMER-ACCOUNT-ID:policy/*"
                }
            }
        },
        {
            "Sid": "AllowResilienceHubCreateGrantForAsyncWorkflows",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::CUSTOMER-ACCOUNT-ID:role/CUSTOMER-ROLE"
            },
            "Action": "kms:CreateGrant",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "resiliencehub.REGION.amazonaws.com",
                    "kms:GrantConstraintType": "EncryptionContextSubset"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:resiliencehub:service-arn":
 "arn:aws:resiliencehub:*:CUSTOMER-ACCOUNT-ID:service/*"
                },
                "ForAllValues:StringEquals": {
                    "kms:GrantOperations": [
                        "Encrypt",
                        "Decrypt",
                        "GenerateDataKey"
                    ]
                }
            }
        }
    ]
}
```

The policy statements provide the following permissions:

- **AllowResilienceHubDescribeKey** – Allows Next generation Resilience Hub to validate that your key exists and is a symmetric encryption key when you specify it during service creation.

- **AllowResilienceHubEncryptDecryptForServices** – Allows Next generation Resilience Hub to encrypt and decrypt service-level data (findings, recommendations, assertions, service functions, events, and assessment data) during synchronous API calls. Scoped to your service resources by encryption context.
- **AllowResilienceHubEncryptDecryptForSystems** – Allows Next generation Resilience Hub to encrypt and decrypt system-level data (system descriptions and user journey descriptions) during synchronous API calls. Scoped to your system resources by encryption context.
- **AllowResilienceHubEncryptDecryptForPolicies** – Allows Next generation Resilience Hub to encrypt and decrypt policy-level data (resiliency policy descriptions) during synchronous API calls. Scoped to your policy resources by encryption context.
- **AllowResilienceHubCreateGrantForAsyncWorkflows** – Allows Next generation Resilience Hub to create a grant for asynchronous assessment workflows. The grant is constrained to only the operations needed (Encrypt, Decrypt, GenerateDataKey) and must include an encryption context subset constraint bound to your service ARN.

For more information about specifying permissions in a policy, see the *AWS Key Management Service Developer Guide*.

For more information about troubleshooting key access, see the *AWS Key Management Service Developer Guide*.

Next generation Resilience Hub encryption context

An encryption context is an optional set of key-value pairs that contain additional contextual information about the data.

AWS KMS uses the encryption context as additional authenticated data to support authenticated encryption. When you include an encryption context in a request to encrypt data, AWS KMS binds the encryption context to the encrypted data. To decrypt data, you must include the same encryption context in the request.

Next generation Resilience Hub encryption context

Next generation Resilience Hub uses the following encryption context keys depending on the resource type:

Encryption context keys

Encryption context key	Scope	Used for
`aws:resiliencehub:service-arn`	Service	Findings, recommendations, assertions, service functions , dependencies, events, and assessment data
`aws:resiliencehub:system-arn`	System	System descriptions and user journey descriptions
`aws:resiliencehub:policy-arn`	Policy	Resiliency policy descriptions

Example encryption context for a service-level operation:

```
"encryptionContext": {
    "aws:resiliencehub:service-arn": "arn:aws:resiliencehub:us-
west-2:111122223333:service/my-service:abc123"
}
```

Using encryption context for monitoring

When you use a symmetric encryption customer managed key to encrypt your data, you can use the encryption context in audit records and logs to identify how the customer managed key is being used. The encryption context appears in logs generated by AWS CloudTrail.

Using encryption context to control access

You can use the encryption context in key policies and IAM policies as `conditions` to control access to your symmetric encryption customer managed key. You can also use encryption context constraints in a grant.

Next generation Resilience Hub uses an encryption context subset constraint in grants to ensure that asynchronous workflows can only encrypt and decrypt data belonging to the specific service the grant was created for.

Monitoring your encryption keys for Next generation Resilience Hub

When you use a customer managed key with your Next generation Resilience Hub resources, you can use AWS CloudTrail to track requests that Next generation Resilience Hub sends to AWS KMS.

CreateGrant

When you create a service with a customer managed key, Next generation Resilience Hub sends a `CreateGrant` request on your behalf to enable asynchronous assessment workflows to use your key. The grant is specific to the service and constrained by encryption context. Next generation Resilience Hub uses `RetireGrant` to remove the grant when you delete the service.

The following example event records the `CreateGrant` operation:

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAEXAMPLE:session-name",
        "arn": "arn:aws:sts::111122223333:assumed-role/YourRole/session-name",
        "accountId": "111122223333",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/YourRole",
                "accountId": "111122223333",
                "userName": "YourRole"
            }
        },
        "invokedBy": "resiliencehub.amazonaws.com"
    },
    "eventTime": "2026-01-15T10:07:22Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "resiliencehub.amazonaws.com",
    "userAgent": "resiliencehub.amazonaws.com",
    "requestParameters": {
        "granteePrincipal": "resiliencehub.amazonaws.com",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "retiringPrincipal": "resiliencehub.amazonaws.com",
```

```
        "operations": [
            "Decrypt",
            "GenerateDataKey",
            "Encrypt"
        ],
        "constraints": {
            "encryptionContextSubset": {
                "aws:resiliencehub:service-arn": "arn:aws:resiliencehub:us-
west-2:111122223333:service/my-service:abc123"
            }
        }
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "56d4e434-abb6-4dd7-8558-ad38560d03b1",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


GenerateDataKey

When Next generation Resilience Hub encrypts data using your customer managed key, it sends a `GenerateDataKey` request to generate a data key. This occurs during both synchronous API calls (such as creating a service with a description) and asynchronous assessment workflows.

The following example event records the `GenerateDataKey` operation:

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "resiliencehub.amazonaws.com"
    },
    "eventTime": "2026-01-15T11:18:36Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "resiliencehub.amazonaws.com",
    "userAgent": "resiliencehub.amazonaws.com",
    "requestParameters": {
        "numberOfBytes": 32,
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "aws:resiliencehub:service-arn": "arn:aws:resiliencehub:us-
west-2:111122223333:service/my-service:abc123",
            "aws-crypto-public-key": "AwAnnorjRE
+DFQYIuDKjGEvlXwro5Rdiegk8flmq7m0N..."
        }
    },
    "responseElements": null,
    "requestID": "c5bedc9b-e6d6-45f8-b121-c9851a3d718a",
    "eventID": "e839a7ed-e4a9-32a3-b92a-2c7237a40c82",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Decrypt

When you retrieve resources through API operations or when assessment workflows process previously stored data, Next generation Resilience Hub sends `Decrypt` requests to decrypt the data.

The following example event records the `Decrypt` operation:

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAEXAMPLE:session-name",
        "arn": "arn:aws:sts::111122223333:assumed-role/YourRole/session-name",
        "accountId": "111122223333",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/YourRole",
                "accountId": "111122223333",
                "userName": "YourRole"
            }
        },
        "invokedBy": "resiliencehub.amazonaws.com"
    },
    "eventTime": "2026-01-15T11:27:49Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "resiliencehub.amazonaws.com",
    "userAgent": "resiliencehub.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "encryptionContext": {
            "aws:resiliencehub:service-arn": "arn:aws:resiliencehub:us-
west-2:111122223333:service/my-service:abc123",
            "aws-crypto-public-key": "A/9P3BC05WjeQONZR1fBiEqWKEse/
Yk1lMxd2VIh2ED5..."
        }
    },
```

```
    "responseElements": null,
    "requestID": "30f8e9bc-4e0a-4359-8bc3-8278ef42c206",
    "eventID": "195ef070-c952-4c28-9883-29bca297a08c",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

DescribeKey

Next generation Resilience Hub sends `DescribeKey` requests to verify that the customer managed key associated with your service exists in the account and region and is a valid symmetric encryption key.

The following example event records the `DescribeKey` operation:

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAEXAMPLE:session-name",
        "arn": "arn:aws:sts::111122223333:assumed-role/YourRole/session-name",
        "accountId": "111122223333",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/YourRole",
                "accountId": "111122223333",
                "userName": "YourRole"
            }
        },
        "invokedBy": "resiliencehub.amazonaws.com"
```

```
    },
    "eventTime": "2026-01-15T10:07:13Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "resiliencehub.amazonaws.com",
    "userAgent": "resiliencehub.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "e427932c-448b-49aa-88e1-b311c27ba753",
    "eventID": "48c596a5-83c7-4603-b0cf-be0ff2548623",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


Learn more

The following resources provide more information about data encryption at rest.

- For more information about AWS Key Management Service basic concepts, see the *AWS Key Management Service Developer Guide*.
- For more information about security best practices for AWS Key Management Service, see the *AWS Key Management Service Developer Guide*.

Encryption in transit

All data in transit is encrypted using TLS 1.2 or later, including:

- API calls to Next generation Resilience Hub endpoints
- Cross-service communication (Next generation Resilience Hub to topology service, Next generation Resilience Hub to Amazon Bedrock)
- Cross-account credential passing (encrypted with AWS KMS before transmission)

VPC endpoints

You can create a VPC endpoint for Next generation Resilience Hub to keep traffic between your VPC and the service within the AWS network, without traversing the public internet. This uses AWS PrivateLink for private connectivity.

You can use VPC endpoints to privately connect your VPC to the next generation of Resilience Hub without requiring traffic to traverse the public internet.

CloudTrail integration

Next generation Resilience Hub integrates with AWS CloudTrail (CloudTrail) to log all API calls made to the service, including calls from the Next generation Resilience Hub console and programmatic calls to the Next generation Resilience Hub API.

Compliance considerations

For information about compliance programs that apply to the next generation of Resilience Hub, see AWS Services in Scope by Compliance Program.

Data retention

The following table shows retention periods for data stored by Next generation Resilience Hub.

Data type	Retention
Assessment results and failure mode findings	2 years
System and service event logs	2 years
Discovered resources	1 year TTL
Topology snapshots	2 years

Data type	Retention
Dependency data	Duration of enablement + 30-day lookback

Generative AI data handling

Failure mode assessments use Amazon Bedrock for AI inference. The following data handling practices apply:

- Customer data is processed in the same AWS Region as the service
- Data is not used to train or improve foundation models
- Assessment inputs and outputs are stored in Next generation Resilience Hub-owned S3 buckets encrypted at rest

Least privilege recommendations

Follow these recommendations to apply least privilege principles to your Next generation Resilience Hub configuration:

1. **Use ExternalId for cross-account roles** – The `ExternalId` condition in cross-account trust policies prevents confused deputy attacks.
2. **Use Organizations Service-Linked Roles** – Avoid manual cross-account role setup when possible. Service-Linked Roles provide automatically scoped, auditable access.

Monitoring Next generation Resilience Hub

You can monitor Next generation Resilience Hub using Amazon CloudWatch, AWS CloudTrail, and Amazon EventBridge to track assessment activity, detect issues, and automate responses.

Topics

- CloudWatch metrics reference
- CloudTrail event logging
- EventBridge notifications
- Setting up CloudWatch alarms for Next generation Resilience Hub

CloudWatch metrics reference

When a failure mode assessment completes successfully, Next generation Resilience Hub publishes policy achievability metrics to your account's Amazon CloudWatch (CloudWatch) under the `ResilienceHub` namespace. Metrics are emitted only upon assessment completion. If no assessment has run, or if the assessment did not produce achievability results, no metrics are reported.

Your service's permission model must include an invoker role with the `cloudwatch:PutMetricData` permission for Next generation Resilience Hub to emit metrics to your account.

Assessment metrics

The following metrics are emitted after each successful failure mode assessment. One data point is published per policy component that has assessment results.

Metric	Dimensions	Description	Values
`PolicyAchievable`	`Service`, `PolicyComponent=AvailabilitySlo`	Whether your availability SLO target is achievable.	1.0 or 0.0
`PolicyAchievable`	`Service`, `PolicyComponent=MultiAzRtoRpo`	Whether your multi-AZ RTO and RPO targets are achievable.	1.0 or 0.0
`PolicyAchievable`	`Service`, `PolicyComponent=MultiRegionRtoRpo`	Whether your multi-Region RTO and RPO targets are achievable.	1.0 or 0.0

The following table describes the metric dimensions.

Dimension	Description
`Service`	The name of the service being assessed.
`PolicyComponent`	The resilience policy component being evaluated. Possible values: `AvailabilitySlo` , `MultiAzRtoRpo` , `MultiRegionRtoRpo` .

> **Note**
>
> Only policy components for which the assessment produces an achievability result are emitted.

Use the `Minimum` statistic when creating alarms on this metric. A `Minimum` of 0.0 indicates that at least one assessment found the policy not achievable during the evaluation period.

CloudTrail event logging

Next generation Resilience Hub is integrated with AWS CloudTrail (CloudTrail). All API calls are logged as events in CloudTrail, including calls from the Next generation Resilience Hub console and programmatic calls to the Next generation Resilience Hub API.

Supported Next generation Resilience Hub events

All Next generation Resilience Hub API actions are logged in CloudTrail, including the following.

Category	Example events
Systems	`CreateSystem` , `DeleteSystem` , `GetSystem` , `ListSystems`
Services	`CreateService` , `UpdateService` , `DeleteService` , `ListServices`
Policies	`CreateResiliencePolicy` , `UpdateResiliencePolicy` , `DeleteResiliencePolicy`

Category	Example events
Assessments	`StartFailureModeAssessment` , `GetFailureModeAssessment` , `ListFailureModeFindings`
Discovery	`StartServiceTopologyDiscovery` , `ListDependencies` , `ClassifyDependency`
Resilience testing	`CreateTest` , `GetTest`, `ListTests` , `UpdateTest` , `DeleteTest` , `StartTestRun` , `StopTestRun` , `GetTestRun` , `ListTestRuns` , `ListTestTemplates` , `GetTestTemplate` , `PutTestSources` , `DeleteTestSources` , `ListTestSources` , `ListTestRunSources` , `ListTestRunEvents` , `ListResolvedTestRunTargetResources` , `ListTestRunSourceEvents` , `ListTestRunDependencies`

Example CloudTrail event

The following is an example CloudTrail event for a `StartFailureModeAssessment` API call.

```
{
  "eventVersion": "1.08",
  "eventSource": "resiliencehub.amazonaws.com",
  "eventName": "StartFailureModeAssessment",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.1",
  "userAgent": "aws-cli/2.x",
  "requestParameters": {
    "serviceArn": "arn:aws:resiliencehub:us-east-1:123456789012:service/
checkout:abc123"
  },
  "responseElements": {
    "assessmentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "status": "PENDING"
  }
}
```

EventBridge notifications

With Amazon EventBridge, you can monitor resources in the next generation of Resilience Hub using event-driven rules. These rules can trigger actions in other AWS services. For example, you can create a rule that signals an Amazon SNS topic whenever a failure mode assessment completes or a new dependency is discovered.

You can create rules in EventBridge to act on the following events from the next generation of Resilience Hub:

- **Failure Mode Assessment Completed** – Emitted when a failure mode assessment completes successfully.
- **Failure Mode Assessment Failed** – Emitted when a failure mode assessment workflow fails.
- **Failure Mode Assessment Queued** – Emitted when an assessment is queued for delayed processing. The assessment is retried within 24 hours.
- **Failure Mode Finding Resolved** – Emitted when a failure mode finding is marked as Resolved or Irrelevant.
- **New Dependency Discovered** – Emitted when dependency discovery identifies a new dependency for your service.

To capture specific events from the next generation of Resilience Hub that you're interested in, define event-specific patterns that EventBridge can use to detect the events. Event patterns have the same structure as the events that they match. The pattern quotes the fields that you want to match and provides the values that you're looking for.

Next generation Resilience Hub emits events on a best-effort basis and delivers them to EventBridge in near real-time when the service is operating normally. However, some situations might delay or prevent event delivery.

For more information about EventBridge rules and event patterns, see Events and event patterns in EventBridge.

Creating an EventBridge rule for events from the next generation of Resilience Hub

With EventBridge, you can create rules that define actions to take when the next generation of Resilience Hub emits events for your services.

Next generation Resilience Hub emits all events with the source `aws.resiliencehub`. Your event pattern must include this source to match events from the next generation of Resilience Hub. You can further filter by `detail-type` to match specific event types.

To enter or paste an event pattern into the EventBridge console, choose the **Enter my own** option. To help you determine event patterns that might be useful, this topic includes Example event patterns and events.

To create a rule for an event from the next generation of Resilience Hub

1. Open the EventBridge console at https://console.aws.amazon.com/events/.
2. For the AWS Region, choose the Region where your service in the next generation of Resilience Hub is deployed.
3. Choose **Create rule**.
4. Enter a **Name** for the rule, and, optionally, a description.
5. For **Event bus**, leave the default value, **default**.
6. Choose **Next**.
7. For **Event source**, leave the default value, **AWS events**.
8. Under **Event pattern**, choose **Enter my own**.
9. Enter or paste an event pattern that includes `"source": ["aws.resiliencehub"]` and the `detail-type` you want to match. For examples, see Example event patterns and events.
10. Choose **Next** and configure your target (for example, an Amazon SNS topic, Lambda function, or CloudWatch log group).
11. Complete the rule creation workflow by choosing **Create rule**.

Example event patterns and events

Event patterns have the same structure as the events that they match. The pattern quotes the fields that you want to match and provides the values that you're looking for.

You can copy and paste event patterns from this section into EventBridge to create rules that monitor events from the next generation of Resilience Hub.

Select all failure mode assessment completion events

The following pattern matches all completed assessments.

```
{
  "source": ["aws.resiliencehub"],
  "detail-type": ["Failure Mode Assessment Completed"]
}
```

Select all failure mode assessment failure events

The following pattern matches all failed assessments.

```
{
  "source": ["aws.resiliencehub"],
  "detail-type": ["Failure Mode Assessment Failed"]
}
```

Select all new dependency discovered events

The following pattern matches all new dependency events.

```
{
  "source": ["aws.resiliencehub"],
  "detail-type": ["New Dependency Discovered"]
}
```

Select all events from the next generation of Resilience Hub

The following pattern matches all events regardless of type.

```
{
  "source": ["aws.resiliencehub"]
}
```

Select assessment events with high-severity findings

The following pattern matches assessments that identified at least one high-severity finding.

```
{
  "source": ["aws.resiliencehub"],
  "detail-type": ["Failure Mode Assessment Completed"],
  "detail": {
    "highSeverityCount": [{"numeric": [">", 0]}]
  }
}
```

The following sections provide example events for each event type.

Failure mode assessment completed event

The following is an example event emitted when a failure mode assessment completes successfully.

```
{
  "version": "0",
  "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "detail-type": "Failure Mode Assessment Completed",
  "source": "aws.resiliencehub",
  "account": "111122223333",
  "time": "2026-01-15T10:30:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:resiliencehub:us-east-1:111122223333:service/my-service:abc123"
  ],
  "detail": {
    "assessmentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "status": "SUCCESS",
    "findingsCount": 5,
    "highSeverityCount": 2
  }
}
```

The `detail` object contains the following fields:

Field	Description
`assessmentId`	The unique identifier of the completed assessment.
`status`	The assessment status. Value: SUCCESS.
`findingsCount`	The total number of failure mode findings identified.
`highSeverityCount`	The number of high-severity findings identified.

Failure mode assessment failed event

The following is an example event emitted when a failure mode assessment fails.

```
{
```

```
  "version": "0",
  "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "detail-type": "Failure Mode Assessment Failed",
  "source": "aws.resiliencehub",
  "account": "111122223333",
  "time": "2026-01-15T10:30:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:resiliencehub:us-east-1:111122223333:service/my-service:abc123"
  ],
  "detail": {
    "assessmentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "status": "FAILED",
    "failureReason": "Unable to assume the invoker role specified in the service
 configuration."
  }
}
```

The `detail` object contains the following fields:

Field	Description
`assessmentId`	The unique identifier of the failed assessment.
`status`	The assessment status. Value: `FAILED`.
`failureReason`	A description of why the assessment failed.

Failure mode assessment queued event

The following is an example event emitted when an assessment is queued for delayed processing. The assessment is retried and completes within 24 hours.

```
{
  "version": "0",
  "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "detail-type": "Failure Mode Assessment Queued",
  "source": "aws.resiliencehub",
  "account": "111122223333",
  "time": "2026-01-15T10:30:00Z",
  "region": "us-east-1",
```

```
  "resources": [
    "arn:aws:resiliencehub:us-east-1:111122223333:service/my-service:abc123"
  ],
  "detail": {
    "assessmentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
  }
}
```

The `detail` object contains the following fields:

Field	Description
`assessmentId`	The unique identifier of the queued assessment.

When the queued assessment completes, a separate `Failure Mode Assessment Completed` or `Failure Mode Assessment Failed` event is emitted.

Failure mode finding resolved event

The following is an example event emitted when a failure mode finding is marked as Resolved or Irrelevant.

```
{
  "version": "0",
  "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "detail-type": "Failure Mode Finding Resolved",
  "source": "aws.resiliencehub",
  "account": "111122223333",
  "time": "2026-01-15T10:30:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:resiliencehub:us-east-1:111122223333:service/my-service:abc123"
  ],
  "detail": {
    "findingId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "status": "RESOLVED",
    "severity": "HIGH",
    "category": "SINGLE_POINT_OF_FAILURE"
  }
}
```

The `detail` object contains the following fields:

Field	Description
`findingId`	The unique identifier of the resolved finding.
`status`	The resolution status. Values: `RESOLVED`, `IRRELEVANT` .
`severity`	The severity of the finding. Values: `HIGH`, `MEDIUM`, `LOW`.
`category`	The failure mode category (for example, `SINGLE_POINT_OF_FAILURE` , `SHARED_FATE`).

New dependency discovered event

The following is an example event emitted when dependency discovery identifies a new dependency for your service. This event is emitted after a stabilization period to reduce noise during initial discovery.

```
{
  "version": "0",
  "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "detail-type": "New Dependency Discovered",
  "source": "aws.resiliencehub",
  "account": "111122223333",
  "time": "2026-01-15T10:30:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:resiliencehub:us-east-1:111122223333:service/my-service:abc123"
  ],
  "detail": {
    "dependencyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "dependencyName": "S3",
    "dnsName": "s3.us-east-1.amazonaws.com.",
    "location": "us-east-1",
    "provider": "AWS",
    "sourceRegions": [
      "us-east-1",
      "us-west-2"
    ]
  }
```

```
}
```

The `detail` object contains the following fields:

Field	Description
`dependencyId`	The unique identifier of the discovered dependency.
`dependencyName`	The name of the dependency (for example, the AWS service name).
`dnsName`	The DNS name used by your service to communicate with the dependency.
`location`	The AWS Region or location where the dependency is hosted.
`provider`	The dependency provider. Values include `AWS`, `ThirdParty` , `Internal`.
`sourceRegions`	The Regions from which your service communicates with this dependency.

Setting up CloudWatch alarms for Next generation Resilience Hub

You can create CloudWatch alarms based on Next generation Resilience Hub metrics to be notified when a resilience policy is no longer achievable.

Alarm: Policy not achievable

The following example creates an alarm that triggers when the availability SLO policy is not achievable for a service named `my-service`.

```
aws cloudwatch put-metric-alarm \
  --alarm-name "ResilienceHub-PolicyNotAchievable-AvailabilitySlo" \
  --metric-name "PolicyAchievable" \
  --namespace "ResilienceHub" \
  --dimensions Name=Service,Value=my-service Name=PolicyComponent,Value=AvailabilitySlo
 \
```

```
  --statistic Minimum \
  --period 86400 \
  --threshold 1 \
  --comparison-operator LessThanThreshold \
  --evaluation-periods 1 \
  --treat-missing-data notBreaching \
  --alarm-actions "arn:aws:sns:us-east-1:123456789012:resilience-hub-alerts"
```

This alarm enters the `ALARM` state when any assessment in the evaluation period reports that the policy is not achievable (metric value 0.0). The `--treat-missing-data notBreaching` setting ensures the alarm does not trigger between assessments when no data points are present.

Quotas and limits

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is AWS Region-specific. You can request increases for some quotas, while other quotas cannot be increased.

Service quotas

The following table lists the quotas for Next generation Resilience Hub.

#	Resource	Default quota	Adjustable
1	Systems per account per AWS Region	500	Yes
2	Services per account per AWS Region	100	Yes
3	User journeys per system	20	No
4	Assessable Resources per service	500	Yes
5	Total Resources per service	20,000	No
6	Input sources per service	20	Yes
7	Service functions per service	20	No
8	Resilience policies per account per AWS Region	100	No

#	Resource	Default quota	Adjustable
9	Failure mode assessments included per service per month	2	No
10	Dependencies tracked per service	10,000	No
11	Cross-account role ARNs per service	5	No
12	Service metrics per service	10	No
13	Active test runs per service	1	No
14	Test sources per test	5	No
15	Tests per test template	1	No
16	Assertions per service (AI-generated and user-created combined)	20	No

Resilience tests run on AWS Fault Injection Service, so AWS FIS service quotas also apply to your test runs. These include quotas for the maximum number of active experiments and the maximum action duration. For more information, see Quotas and limitations for AWS Fault Injection Service.

Hard limits

The following quotas cannot be increased.

#	Resource	Hard limit
1	Systems per account per AWS Region (maximum)	1,000
2	Input sources per service (maximum)	100
3	Assessable resources per service (maximum)	1,000
4	User journeys per system (maximum)	100
5	Cross-account role ARNs per service	5

#	Resource	Hard limit
6	Failure mode assessments included per service per month	2
7	Assertions per service (maximum)	20

Requesting quota increases

You can request an increase for adjustable quotas using the Service Quotas console or by contacting AWS Support.

To request a quota increase using Service Quotas:

1. Open the Service Quotas console.
2. In the navigation pane, choose **AWS services**.
3. Choose **AWS Resilience Hub**.
4. Choose the quota that you want to increase.
5. Choose **Request quota increase**.
6. Enter the new quota value and choose **Request**.

To request a quota increase through AWS Support:

1. Open the AWS Support Center.
2. Choose **Create case**.
3. Choose **Service limit increase**.
4. Select **Resilience Hub** as the service.
5. Fill in the required information and submit your request.

Quota increase requests are typically processed within a few business days.

Troubleshooting Next generation Resilience Hub

This section provides solutions for common issues you might encounter when using the next generation of Resilience Hub.

Topics

- Dependency discovery issues
- Failure mode assessment issues
- IAM and permissions errors
- AWS Organizations configuration issues
- Resource discovery issues
- Resilience testing issues
- Known issues and workarounds

Dependency discovery issues

The following are common issues related to dependency discovery.

Discovery not returning expected dependencies

Symptom: You enabled dependency discovery but don't see dependencies you know exist.

The following table lists possible causes and solutions.

Cause	Solution
Dependency uses direct IP (not DNS)	Dependency discovery relies on DNS queries. Connections made by IP address are not discovered. Manually track IP-based dependencies.
Compute resources not in a VPC	Non-VPC Lambda functions don't generate Route 53 resolver queries. Connect Lambda to a VPC or manually track dependencies.
Dependency not called in 35-day window	Dependencies must have been called within the 35-day lookback period. Very infrequent calls may not appear.
DNS resolution through non-Route 53 resolver	Custom DNS resolvers bypass Route 53 query logging. Ensure your VPC uses the default Route 53 resolver.

Kubernetes attribution gaps

Symptom: Dependencies are discovered but attributed to the wrong service in a shared EKS cluster.

Cause: When multiple services share an EKS cluster, DNS queries are attributed at the node level, not the pod level.

Solution: Enhanced pod-level attribution is planned for a future release. Currently, review dependencies manually for shared clusters.

Non-VPC Lambda not appearing

Symptom: Lambda function dependencies are not discovered.

Cause: Lambda functions not connected to a VPC do not make DNS queries through Route 53 resolvers.

Solution: Either connect the Lambda function to a VPC, or manually track its dependencies outside of Resilience Hub.

Failure mode assessment issues

The following are common issues related to failure mode assessments.

Assessment not completing

Symptom: Assessment stays in `IN_PROGRESS` status for more than 30 minutes.

The following table lists possible causes and solutions.

Cause	Solution
Large service (many resources)	Assessments for services with 1,000 or more resources may take longer. Wait up to 60 minutes.
Invoker role permissions issue	Verify the invoker role has `ReadOnlyAccess` and `AWSResilienceHubV2AssessmentExecutionPolicy` attached.

Cause	Solution
Topology not completed	Ensure `StartServiceTopologyDiscovery` completed successfully before starting an assessmen t.
Service error	If the assessment fails, check the error message in the `GetFailureModeAssessment` response.

Assessment fails with topology not found

Symptom: `StartFailureModeAssessment` returns an error about missing topology.

Solution: Run `StartServiceTopologyDiscovery` first and wait for it to complete successfully. Assessments require a completed topology.

Failure mode findings not matching expected policy

Symptom: Assessment failure mode findings don't seem to relate to your applied policy.

Possible causes:

- Policy was applied after the assessment ran – re-run the assessment.
- Multiple policies apply (from user journey and service level) – check all applied policies.
- The failure mode finding relates to a Well-Architected best practice, not a specific policy requirement.

Assessment produces no failure mode findings

Symptom: Assessment completes successfully but returns zero failure mode findings.

Possible causes:

- Service has very few resources (assessment needs sufficient architecture to analyze).
- No policy is applied (assessments without policies produce fewer findings).
- Architecture already meets all requirements.

IAM and permissions errors

The following are common IAM and permissions errors.

Common permission error codes and resolution

The following table lists common permission error codes and their recommended resolution.

Error	Cause	Resolution
`AccessDeniedException: Unable to assume role`	Invoker role trust policy doesn't include `resiliencehub.amazonaws.com`	Update the role's trust policy.
`AccessDeniedException: Cross-account role`	Cross-account role trust policy doesn't allow assumption from the invoker role	Verify the trust policy and `ExternalId` .
`AccessDeniedException: sts:AssumeRole`	Invoker role lacks `sts:AssumeRole` permission for cross-account roles	Add `sts:AssumeRole` permission to the invoker role.
`InvalidParameterException: Role does not exist`	Role name in `permissionModel` doesn't match an existing IAM role	Verify the role name and account.

Verifying role configuration

Check that your invoker role is correctly configured using the following AWS CLI commands.

To verify that the role exists:

```
aws iam get-role --role-name AWSResilienceHubAssessmentRole
```

To verify the trust policy:

```
aws iam get-role --role-name AWSResilienceHubAssessmentRole \
  --query 'Role.AssumeRolePolicyDocument'
```

To simulate whether a principal can perform required actions:

```
aws iam simulate-principal-policy \
  --policy-source-arn arn:aws:iam::123456789012:role/AWSResilienceHubAssessmentRole \
  --action-names resiliencehub:GetService resiliencehub:StartFailureModeAssessment
```

AWS Organizations configuration issues

The following are common AWS Organizations configuration issues.

Delegated administrator setup errors

Error	Cause	Resolution
`AWSOrganizationsNotInUseException`	Organizations not enabled	Enable AWS Organizations with all features.
`AccountNotRegisteredException`	Account is not a member of the organization	Verify account membership.
`ConstraintViolationException`	Service trust not enabled	Run `enable-aws-service-access` first.

Member account visibility gaps

Symptom: The delegated administrator cannot see services in a member account.

The following table lists possible causes and solutions.

Cause	Solution
Service-linked role not yet created	For large organizations, service-linked role creation may take time. Wait and check again.

Cause	Solution
Account was suspended during setup	Unsuspend the account. The 12-hour reconciliation job will create the service-linked role.
Account recently joined organization	New accounts receive service-linked roles automatic ally, but there may be a brief delay.
Service trust disabled	Re-enable trusted access from the management account.

Verifying the service-linked role exists in a member account

From the member account, run the following command to verify the service-linked role exists:

```
aws iam get-role --role-name AWSServiceRoleForResilienceHub
```

If the role doesn't exist, verify that trusted access is enabled and wait for the reconciliation job, which runs every 12 hours.

Resource discovery issues

The following are common resource discovery issues.

CloudFormation stack not found

Symptom: Resource discovery fails with "stack not found."

Solutions:

- Verify the stack ARN is correct and the stack exists.
- Verify the invoker role has `cloudformation:DescribeStackResources` permission.
- For cross-account stacks, verify the cross-account role has access.

Terraform state file not accessible

Symptom: Resource discovery fails reading the Terraform state file.

Solutions:

- Verify the Amazon S3 bucket and key path are correct.
- Verify the invoker role has `s3:GetObject` permission on the state file.
- Verify the state file is in a supported format.

Resources discovered but topology generation fails

Symptom: Resources appear in the list but topology shows no connections.

Solutions:

- Verify the invoker role has `ReadOnlyAccess` (required for topology queries).
- Check that resources are in a VPC (topology generation maps VPC-based connections).
- For EKS resources, verify Kubernetes RBAC is configured.

Unsupported and excluded resource types

Next generation Resilience Hub excludes certain resource types from discovery because they have no independent resilience value. If you expect a resource to appear in your discovered resources but it does not, it might be an excluded type.

Resource types never discovered

Next generation Resilience Hub never stores the following resource types during discovery. These resource types include auxiliary infrastructure, point-in-time backups, and operational tooling with no bearing on resilience analysis.

Resource type
`AWS::EC2::EIP`
`AWS::EC2::Volume`
`AWS::EC2::Snapshot`
`AWS::FSx::Snapshot`
`AWS::Lightsail::DiskSnapshot`
`AWS::RDS::DBSnapshot`

Resource type
AWS::RDS::DBClusterSnapshot
AWS::EC2::FlowLog
AWS::CloudWatch::Dashboard
AWS::Logs::LogGroup
AWS::Logs::MetricFilter
AWS::SSM::ResourceDataSync
AWS::RDS::DBParameterGroup
AWS::RDS::DBClusterParameterGroup
AWS::ElastiCache::ParameterGroup
AWS::RDS::DBSubnetGroup
AWS::ElastiCache::SubnetGroup
AWS::EC2::SubnetNetworkAclAssociation
AWS::EC2::VPCGatewayAttachment
AWS::CloudFormation::Stack
AWS::CloudFormation::WaitConditionHandle
AWS::CodeDeploy::DeploymentGroup
AWS::KMS::Alias
AWS::SecretsManager::SecretTargetAttachment
AWS::CloudFormation::CustomResource

Resource types stored but not billed

Next generation Resilience Hub discovers and stores the following resource types for context, but does not count them toward your billable resource quota. They do not appear in resilience analysis.

Resource type
AWS::IAM::Role
AWS::IAM::Policy
AWS::IAM::ManagedPolicy
AWS::IAM::InstanceProfile
AWS::Lambda::LayerVersion
AWS::Lambda::Permission
AWS::EC2::LaunchTemplate
AWS::EC2::SecurityGroup
AWS::EC2::SubnetRouteTableAssociation
AWS::CloudFront::OriginRequestPolicy
AWS::CloudFront::CloudFrontOriginAccessIdentity
AWS::SecretsManager::Secret
AWS::CloudWatch::Alarm
AWS::CDK::Metadata

Difference between discovered resources and topology resources

You might see more resources in the `ListResources` API response than appear in your topology diagram. Not all discovered resources appear in the topology view. The topology is a connectivity graph that shows how resources interact. Next generation Resilience Hub removes resources from the topology view that have no connections to other resources.

EC2 instances or ECS tasks not appearing as billable resources

When a parent resource (such as an Auto Scaling group, ECS service, or EKS deployment) manages an EC2 instance, ECS task, EKS pod, or network interface, Next generation Resilience Hub classifies that resource as *ephemeral*. Ephemeral resources are transient. The parent resource replaces them automatically through lifecycle events. Next generation Resilience Hub assesses them through their parent, not independently.

The following resource types are ephemeral when managed by a parent:

Resource type	Example parent
`AWS::EC2::Instance`	Auto Scaling group or EKS node group
`AWS::ECS::Task`	ECS service
`AWS::EKS::Pod`	EKS deployment or replica set
`AWS::EC2::NetworkInterface`	EC2 instance managed by Auto Scaling

Standalone instances of these types (not managed by a parent) remain billable.

Resilience testing issues

The following are common issues related to resilience testing.

Test run fails to start

Symptom: Starting a test run returns a `ConflictException`.

Cause: Next generation Resilience Hub runs a single test run at a time for a target. The previous run for the service is still in progress.

Solution: Wait for the current run to complete, or stop it with `StopTestRun`, and then start the new run.

Test run fails with an access denied error

Symptom: A test run fails with an `AccessDenied` error from AWS FIS or one of the targeted services.

The following table lists possible causes and solutions.

Cause	Solution
Execution role trust policy doesn't allow AWS FIS	Verify the role trusts `fis.amazonaws.com` and that the `aws:SourceAccount` and `aws:SourceArn` conditions match your account. For the trust policy, see IAM execution roles for resilience testing.
Execution role missing permissions for the test template	Each test template runs a different set of AWS FIS actions. Attach the permissions policy for the template you are running. For the policies, see IAM execution roles for resilience testing.
Multi-account role chain not configured	For a multi-account test, verify the orchestrator role can assume the target role in each account that contains targeted resources.

Test run stops before completing

Symptom: A test run stops on its own before it finishes.

Cause: A CloudWatch alarm that you configured as a stop condition breached its threshold, and resilience testing stopped the run to limit impact.

Solution: Review the alarm that triggered the stop condition to determine the impact on your service. After you address the underlying issue, start a new test run.

Test run completes with all actions skipped

Symptom: A test run completes but no faults were injected. All actions show `skipped`.

Cause: No resources match any action's target resource type in the selected Availability Zone or Region.

Solution: Verify resource discovery has completed and you selected the correct Availability Zone or Region.

Dependency fault action fails with setup error

Symptom: A dependency fault action fails because the required agent or sidecar is not configured.

Cause: Packet loss actions require SSM Agent on Amazon EC2, an SSM container in Amazon ECS task definitions, or a Kubernetes service account for Amazon EKS pods.

Solution: Follow the setup steps for your compute type in the AWS FIS actions reference. For Amazon EKS pods, the service account must exist in every targeted namespace, and the execution role must be mapped to the cluster from the account that contains the cluster. Complete the **(Optional) Resilience testing** parts of the Amazon EKS permissions steps. For more information, see Required IAM permissions and roles.

Not all dependencies visible in the test

Symptom: Not all discovered dependencies are shown in the dropdown.

Cause: Dependencies are filtered per test to show only those relevant to the test's scope.

Solution: Each test shows a different subset. See the individual test pages for which dependencies are shown.

Known issues and workarounds

This section lists known issues in the next generation of Resilience Hub and their recommended workarounds. For the most up-to-date information, see the Next generation Resilience Hub release notes.

API reference

This section provides an overview of the next generation of Resilience Hub API operations, organized by feature area. For complete API specifications including request and response schemas, see the Next generation Resilience Hub API Reference.

Topics

- Making API requests and authentication
- Actions
- Data types

- Error codes

Making API requests and authentication

Endpoint

```
https://resiliencehub.{region}.amazonaws.com
```

API version

Next generation Resilience Hub APIs use the `/v2` path prefix. All requests must be signed with AWS Signature Version 4 (SigV4).

Authentication

All API calls require valid AWS credentials. Next generation Resilience Hub uses AWS IAM for authorization. Ensure your IAM policy grants the necessary `resiliencehub:*` actions.

Actions

the next generation of Resilience Hub API provides the following categories of actions.

Resilience policies

Action	Method	Description
`CreatePolicy`	POST	Create a resilience policy with availability SLO, RTO/RPO targets, and DR approach.
`UpdatePolicy`	POST	Update policy targets and configuration.
`GetPolicy`	GET	Retrieve policy details.
`ListPolicies`	GET	List policies with associated service count.
`DeletePolicy`	POST	Delete a policy.
`ListPolicyEvents`	GET	List the events in a policy's timeline.

Systems

Action	Method	Description
`CreateSystem`	POST	Create a system with optional dependency discovery enablement.
`UpdateSystem`	POST	Update system description, dependency discovery, and KMS key.
`GetSystem`	GET	Retrieve system details.
`ListSystems`	GET	List systems, filterable by `organizationId` , `ouId`, and `accountId` .
`DeleteSystem`	POST	Delete a system (must have no associated services).

User journeys

Action	Method	Description
`CreateUserJourney`	POST	Create a user journey under a system.
`UpdateUserJourney`	POST	Update user journey services and policy.
`GetUserJourney`	GET	Retrieve user journey details.
`ListUserJourneys`	GET	List user journeys for a system.
`DeleteUserJourney`	POST	Delete a user journey.

Services

Action	Method	Description
`CreateService`	POST	Create a service with regions, permission model, report configuration, and dependency discovery settings.
`UpdateService`	POST	Update service configuration including permission model and dependency discovery.
`GetService`	GET	Retrieve full service details including effective policy values and resilience score.
`ListServices`	GET	List services, filterable by `systemId`, `userJourneyId` , `organizationId` , `ouId`, `accountId` , `assessmentStatus` , and `policyArn` .
`DeleteService`	POST	Delete a service.

Input sources

Action	Method	Description
`CreateInputSource`	POST	Configure a resource discovery source (resource tags, CloudFormation stack, Terraform state file, EKS cluster, or design file). For a single tag-based input source with multiple tags, the service discovers only resources that match all specified tags. For multiple tag-based input sources, the service discovers resources that match any of them.
`ListInputSources`	GET	List input sources for a service.
`DeleteInputSource`	POST	Delete an input source.

Resources

Action	Method	Description
`ListResources`	GET	List discovered resources for a service, filterable by `serviceFunctionId` and `awsRegion` .

Failure mode assessments

Action	Method	Description
`StartFailureModeAssessment`	POST	Trigger an asynchronous AI-powered assessment that creates topology, service functions, and failure mode findings.
`ListFailureModeAssessments`	GET	List assessments with status, resilience score, and report info.

Resilience testing

Action	Method	Description
`ListTestTemplates`	GET	List the available test templates.
`GetTestTemplate`	GET	Retrieve a test template, including its actions and parameters.
`CreateTest`	POST	Create a test for a service from a test template.
`GetTest`	GET	Retrieve a test configuration including parameters, stop conditions, and role.
`UpdateTest`	POST	Update a test configuration including parameters, stop conditions, and role.
`ListTests`	GET	Lists tests for a service.

Action	Method	Description
DeleteTest	POST	Delete a test.
StartTestRun	POST	Start a test run for a service.
StopTestRun	POST	Stop a test run.
ListTestRuns	GET	List test runs for a service.
GetTestRun	GET	Retrieve a test run including status, parameters, and timestamp.
ListTestRunEvents	GET	List the events in a test run's timeline.
PutTestSources	POST	Add or update success criteria alarms or observability alarms for a test.
DeleteTestSources	POST	Remove success criteria alarms or observability alarms from a test.
ListTestSources	GET	List success criteria alarms or observability alarms for a test.
ListTestRunSources	GET	List success criteria alarms or observability alarms for a test run.
ListResolvedTestRunTargetResources	GET	List the resources that were resolved as targets for a test run.
ListTestRunSourceEvents	GET	List the state-change events observed for a monitoring source associated with a test run, such as Amazon CloudWatch alarm state changes.
ListTestRunDependencies	GET	List the dependencies a test run blocked, with the classification captured when the run started.

Topology

Action	Method	Description
`ListServiceTopologyEdges`	GET	List resource dependency edges for a service.

Service functions

Action	Method	Description
`CreateServiceFunction`	POST	Create a service function with criticality (`PRIMARY/SUPPLEMENTAL`).
`UpdateServiceFunction`	POST	Update service function properties.
`ListServiceFunctions`	GET	List service functions for a service.
`DeleteServiceFunction`	POST	Delete a service function.
`AddServiceFunctionResources`	POST	Associate resources with a service function (up to 10 per call).
`DeleteServiceFunctionResources`	POST	Remove resource associations (up to 10 per call).

Failure mode findings

Action	Method	Description
`GetFailureModeFinding`	GET	Retrieve failure mode finding details including AI-generated recommendations.

Action	Method	Description
`UpdateFailureModeFinding`	POST	Update failure mode finding status (resolved, irrelevant) and add comments.
`ListFailureModeFindings`	GET	List failure mode findings, filterable by `severity`, `failureCategory` , and `status`.

Assertions

Action	Method	Description
`CreateAssertion`	POST	Create a user-created assertion for a service. Assertion text can be up to 1,000 characters.
`UpdateAssertion`	POST	Update the text of an assertion. Updating an AI-generated assertion changes its source to USER.
`ListAssertions`	GET	List assertions for a service, optionally filtered by source (`AI_GENERATED` or USER).
`DeleteAssertion`	POST	Delete an assertion.

Dependencies

Action	Method	Description
`UpdateDependency`	POST	Update dependency metadata (comment, criticality classification).
`ListDependencies`	GET	List discovered dependencies for a service.

Reports

Action	Method	Description
`CreateReport`	POST	Generate a PDF report (`FAILURE_MODE` , `DEPENDENCY` , or `TESTING` type); writes to customer Amazon S3 bucket.
`ListReports`	GET	List generated reports with status and Amazon S3 output location.

Events

Action	Method	Description
`ListServiceEvents`	GET	List audit events for a service (includes actor: `USER`/`SYSTEM`, score changes).
`ListSystemEvents`	GET	List audit events for a system.

Migration

Action	Method	Description
`ImportApp`	POST	Import a v1 Resilience Hub application into a v2 service.
`ImportPolicy`	POST	Import a v1 policy into v2 format.

Data types

The following table lists the key data types used by the next generation of Resilience Hub API.

Type	Description
`SystemEntity`	System with ARN, name, description, and creation time.
`UserJourneyEntity`	User journey with name, description, service list, and policy.
`ServiceEntity`	Service with ARN, name, system association, permission model, and input sources.
`ResiliencePolicyEntity`	Policy with name and components (DR, availability, performance).
`AssessmentEntity`	Assessment with ID, status, timestamps, and failure mode findings count.
`FindingEntity`	Failure mode finding with ID, name, description, severity, and recommendations.
`RecommendationEntity`	Recommendation with name, description, cost, and complexity.
`DependencyDiscoveryConfig`	Dependency discovery configuration with status, eligible resource count, message, and last-updated timestamp. Returned by `GetService` and `ListServices`.
`DependencyEntity`	Dependency with name, location, criticality, and allowed status.
`TopologyEntity`	Topology with ID, timestamp, resource count, and edges.

DependencyDiscoveryConfig structure

```
{
  "status": "INITIALIZING | ENABLED | DISABLED",
  "updatedAt": "timestamp",
  "eligibleResourceCount": integer,
  "message": "string"
}
```

The following table describes the fields in the `DependencyDiscoveryConfig` structure.

Field	Required	Description
`status`	Yes	The current state of dependency discovery. Valid values: `INITIALIZING` (discovery is running), `ENABLED` (discovery is complete and active), or `DISABLED` (discovery is turned off).
`updatedAt`	No	The timestamp when the dependency discovery status was last updated.
`eligibleResourceCount`	No	The number of compute resources eligible for dependency discovery. Returns `null` until the resource discovery process completes its first run.
`message`	No	A message describing the current state of discovery. Returns `null` when discovery is complete and dependencies are available.

Permission model structure

```
{
  "invokerRoleName": "string",
  "crossAccountRoles": [
    {
      "crossAccountRoleArn": "string",
      "externalId": "string"
    }
  ]
}
```

Policy components structure

```
{
  "multiRegionDr": {
    "required": "boolean",
    "rtoMinutes": "integer",
```

```
    "rpoMinutes": "integer"
  },
  "availabilitySlo": {
    "targetPercentage": "number",
    "performanceConditions": {
      "expression": "string",
      "conditions": [
        {
          "type": "FAULT_RATE | LATENCY | VOLUME",
          "operator": "LESS_THAN | GREATER_THAN",
          "value": "number",
          "percentile": "string"
        }
      ]
    }
  },
  "dataProtection": {
    "rpoMinutes": "integer"
  }
}
```

Error codes

The following table lists error codes returned by the next generation of Resilience Hub API.

Error code	HTTP status	Description
`ValidationException`	400	Request parameters are invalid.
`ResourceNotFoundException`	404	The specified resource does not exist.
`ConflictException`	409	Operation conflicts with current state (for example, an assessment is already running).
`AccessDeniedException`	403	Caller lacks required permissions.
`ThrottlingException`	429	Request rate exceeded.
`InternalServerException`	500	Internal service error.

Error code	HTTP status	Description
`ServiceQuotaExceededExcepti on`	402	A service quota has been exceeded.

Common error scenarios

Scenario	Error	Resolution
Starting assessment without topology	`ConflictException`	Run `StartServiceTopolo gyDiscovery` first.
Starting assessment while one is running	`ConflictException`	Wait for the current assessment to complete.
Deleting system with services	`ConflictException`	Remove all service associations first.
Cross-account access without role	`AccessDen iedException`	Configure cross-account roles or enable Organizations.
Exceeding 5 cross-account roles	`ServiceQu otaExceed edException`	Use AWS Organizations for larger deployments.

Document history for the Next generation Resilience Hub User Guide

The following table describes the documentation releases for the next generation of Resilience Hub.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.