



User Guide

AWS Resilience Hub



AWS Resilience Hub: User Guide

Copyright © 2024 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

What is AWS Resilience Hub?	1
AWS Resilience Hub – Resilience management	1
How AWS Resilience Hub works	2
AWS Resilience Hub – Resilience testing	5
AWS Resilience Hub concepts	6
Resiliency	6
Recovery point objective (RPO)	6
Recovery time objective (RTO)	6
Estimated workload recovery time objective	6
Estimated workload recovery point objective	6
Application	6
Application Component	7
Application compliance status	7
Drift detection	8
Resiliency assessment	8
Resiliency score	8
Disruption type	8
Fault injection experiments	9
SOP	9
Supported AWS Resilience Hub resources	10
Getting started	13
Prerequisites	13
Add an application	14
Step 1: Get started by adding an application	15
Step 2: Manage your application resources	15
Step 3: Add resources to your AWS Resilience Hub application	16
Step 4: Set RTO and RPO	21
Step 5: Setup scheduled assessment and drift notification	22
Step 6: Setup permissions	23
Step 7: Configure the application configuration parameters	25
Step 8: Add tags to your application	25
Step 9: Review and publish	26
Step 10: Run an assessment	26
Using AWS Resilience Hub	27

AWS Resilience Hub dashboard	27
Application status	27
Application resiliency score over time	28
Implemented alarms	28
Implemented experiments	29
Managing applications	29
Viewing application summary	31
Editing application resources	33
Grouping resources in an AppComponent	42
Publish a new application version	45
Viewing application versions	46
Viewing resources of your application	47
Deleting an application	49
Application configuration parameters	49
Managing resiliency policies	50
Creating resiliency policies	51
Accessing resiliency policy details	54
Managing resiliency assessments	56
Running resiliency assessments	56
Reviewing assessments reports	57
Deleting resiliency assessments	66
Managing alarms	67
Creating alarms from the operational recommendations	67
Viewing alarms	70
Managing standard operating procedures	73
Building an SOP based on AWS Resilience Hub recommendations	75
Creating a custom SSM document	76
Using a custom SSM document instead of the default	77
Testing SOPs	77
Viewing standard operating procedures	77
Managing Amazon Fault Injection Service experiments	79
Creating AWS FIS experiments from the operational recommendations	80
Running an AWS FIS experiment from AWS Resilience Hub	82
Viewing fault injection experiments	82
Amazon Fault Injection Service experiment failures/status check	85
Understanding resiliency scores	88

Accessing the Resiliency score of your applications	88
Calculating resiliency scores	91
Integrating recommendations into applications	103
Modifying the AWS CloudFormation template	105
Using AWS Resilience Hub APIs to describe and manage application	109
Preparing the application	109
Create an application	109
Create resiliency policy	110
Import application resource and monitor import status	111
Publish your application and assign resiliency policy	114
Running and analyzing the application	115
Run and monitor a resiliency assessment	116
Create resiliency policy	119
Modify your application	134
Manually add resources	134
Grouping resources into a single Application Component	135
Excluding a resource from an AppComponent	137
Security	139
Data protection	139
Encryption at rest	140
Encryption in transit	140
Identity and Access Management	141
Audience	141
Authenticating with identities	142
Managing access using policies	145
How AWS Resilience Hub works with IAM	148
Setup IAM roles and permissions	161
Troubleshooting	162
AWS Resilience Hub access permissions reference	164
AWS managed policies	177
Importing Terraform state file into AWS Resilience Hub	186
Enabling AWS Resilience Hub access to your Amazon EKS cluster	190
Enabling AWS Resilience Hub to publish to your Amazon SNS topics	202
Limiting permissions to include or exclude AWS Resilience Hub recommendations	203
Infrastructure security	204
Working with other services	205

AWS CloudFormation	205
AWS Resilience Hub and AWS CloudFormation templates	205
Learn more about AWS CloudFormation	206
AWS CloudTrail	206
AWS Systems Manager	206
AWS Trusted Advisor	207
Document history	210
AWS Glossary	234

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 Amazon 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 Route 53 Application Recovery Controller (Route 53 ARC)	What is Amazon Route 53 Application Recovery Controller

Topics

- [AWS Resilience Hub – Resilience management](#)
- [AWS Resilience Hub – Resilience testing](#)
- [AWS Resilience Hub concepts](#)
- [AWS Resilience Hub supported resources](#)

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, AppRegistry 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 CI/CD, 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 AWS Service Catalog AppRegistry.

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 allows you to perform Amazon Fault Injection Service (AWS FIS) tests and experiments on your AWS workloads and maintain optimal resilience. These tests stress an application 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
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 AppComponent 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 AppComponent 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.

Fault injection experiments

AWS Resilience Hub recommends tests 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 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 "AWSResilienceHubAssessmentExecutionPolicy"](#).

AWS Resilience Hub supports resources from the following AWS services:

- Compute
 - Amazon Elastic Compute Cloud (Amazon EC2)
 - 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
- 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 Route 53 Application Recovery Controller (Amazon Route 53 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 Route 53 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 AppComponent](#).

- 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

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 AppRegistry application.

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”](#)
- 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 “AWSResilienceHubAssessmentExecutionPolicy”](#).

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 application, AWS CloudFormation stacks, AWS Resource Groups, or AppRegistry 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

- [Step 1: Get started by adding an application](#)
- [Step 2: How is your application managed?](#)
- [Step 3: Add resources to your AWS Resilience Hub application](#)
- [Step 4: Set RTO and RPO](#)
- [Step 5: Setup scheduled assessments and drift notification](#)
- [Step 6: Setup permissions](#)
- [Step 7: Configure the application configuration parameters](#)

- [Step 8: Add tags](#)
- [Step 9: Review and publish your AWS Resilience Hub application](#)
- [Step 10: Run an assessment of your AWS Resilience Hub application](#)

Step 1: 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

[Step 2: How is your application managed?](#)

Step 2: How is your application managed?

In addition to AWS CloudFormation stacks, AWS Resource Groups, AppRegistry 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, AppRegistry 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 one of the resource collections 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

[Step 3: Add resources to your AWS Resilience Hub application](#)

Step 3: Add resources to your AWS Resilience Hub application

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

- [the section called “Add resource collections”](#)
- [the section called “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 AppRegistry applications
- Using Terraform state files
- Using an existing AWS Resilience Hub application

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 stacks** to discover your stack-based resources.
2. Choose stacks from the **Select 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, enter the Amazon Resource Name (ARN) of the stack in the **Add stack outside of AWS Region** 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 **Select resource groups** dropdown list.

To use AWS Resource Groups that are in a different AWS account, different Region, or both, enter the Amazon Resource Name (ARN) of the stack in the **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 AppRegistry applications

You can add only one AppRegistry application at a time.

Choose the AppRegistry applications 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 **AppRegistry** to select from a list of applications created in AppRegistry.
2. Choose applications, which were created in AppRegistry, from the **Select application** dropdown list. You can choose only one application at a time.

Using Terraform state files

Choose the Terraform state file that contains your 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 URL** field, and choose **Add S3 URL**.

The limit for Terraform state files is 4 megabytes (MB).

3. Select your S3 bucket from the **Buckets** section.
4. From the **Objects** section, select a key, and choose **Choose**.

Using an existing AWS Resilience Hub application

To get started, use an existing application.

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

1. Select **Existing application** to build your application from an existing application.
2. Select an application from the **Select existing application** dropdown list.

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. 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, enter the Amazon Resource Name (ARN) of the stack in the **Cross account or Region** 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

[Step 4: Set RTO and RPO](#)

Step 4: 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 resilience policy** option.
2. Enter a name for the resiliency policy.
3. (Optional) Enter a description for the resiliency policy.
4. Define your RTO/RPO in the **RTO/RPO targets** section.

Note

- We have 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 "Step 5: Setup scheduled assessment and drift notification"](#)

Step 5: 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 `AwsResilienceHubPeriodicAssessmentRole`

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

[Step 6: Setup permissions](#)

Step 6: 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 in **Step 1** and **Step 3** will be ignored.

Next

[Step 7: Configure the application configuration parameters](#)

Step 7: 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

[Step 8: Add tags](#)

Step 8: 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

[Step 9: Review and publish your AWS Resilience Hub application](#)

Step 9: Review and publish your AWS Resilience Hub application

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

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

[Step 10: Run an assessment of your AWS Resilience Hub application](#)

Step 10: 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 dashboard](#)
- [Describing and managing AWS Resilience Hub Applications](#)
- [Managing resiliency policies](#)
- [Running and managing AWS Resilience Hub resiliency assessments](#)
- [Managing alarms](#)
- [Managing standard operating procedures](#)
- [Managing Amazon Fault Injection Service experiments](#)
- [Understanding resiliency scores](#)
- [Integrating operational recommendations into your application with AWS CloudFormation](#)

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 Amazon 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 fault injection 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 AWS 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](#)
- [Grouping resources in an AppComponent](#)
- [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

- [Summary](#)
- [Application resiliency](#)
- [Implemented alarms](#)
- [Implemented experiments](#)

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 on 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 Amazon 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 [Step 5: 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 [Step 5: 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 [Step 6: 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, AppRegistry 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 “Step 3: 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 criterion 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, AppRegistry 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 AppComponent](#).
- **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, AppRegistry 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 [Step 3: Add resources to your AWS Resilience Hub application](#), 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 AppComponent](#).

The **AppComponent** section lists all the logical components that the resources are grouped into. You can identify the AppComponent 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](#).


Grouping resources in an AppComponent

An 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, then both databases belong to the same Application Component (AppComponent). AWS Resilience Hub has rules governing 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`.

When the application is imported into AWS Resilience Hub with AWS CloudFormation stack, Terraform state file, AWS Resource Groups, Amazon Elastic Kubernetes Service cluster, or an AppRegistry application, AWS Resilience Hub makes its best effort to group related resources into the same AppComponent, but might not always be 100 percent accurate. You know the architecture of your application the best, so you can regroup resources that have already been grouped by AWS Resilience Hub into a different AppComponent. For example, if you have three EC2 instances in an AWS CloudFormation stack, AWS Resilience Hub creates a single AppComponent per EC2 instance, but all three EC2 instances might be running the same application software. In this case, the correct choice is to regroup the three EC2 instances under a single ComputeAppComponent. When regrouping resources, you should only regroup the resource to a single AppComponent. You can also expand your resource list and combine ungrouped resources into an AppComponent.

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::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`

The following are examples of correct groupings:

- Group primary databases and replicas under a single `AppComponent`.
- Group an Amazon S3 bucket and its replication under a single `AppComponent`.
- Group Amazon EC2 instances that run the same application under a single `AppComponent`.
- ~~Group an Amazon SQS queue and its dead-letter queue under a single `AppComponent`.~~

- Group Amazon ECS services in one Region and failover Amazon ECS services in another Region under a single AppComponent.

 **Note**

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

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 resource that you want to regroup.
7. From **Actions**, choose **Change AppComponent**.

The **Change AppComponent** dialog box displays.

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.

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. Choose the resources that you want to group.

Note

You cannot choose manually added resources.

7. Choose **Actions**, and then choose **Group resources**.

The **Combine AppComponent** window displays.

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.

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 AWS Resilience Hub resiliency assessments](#).

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 view 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, manually added application, AppRegistry 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 AppComponent](#).
- **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, AppRegistry 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 of ten application limits, 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 "Step 1: Get started by adding an application"](#) through [the section called "Step 8: 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**.

- (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.

- (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*.
- To create the policy, choose **Create**.

To create resiliency policies in Resiliency policies

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

The **Create resiliency policy** page displays.

- Enter a name for the policy.
- (Optional) Enter a description for the policy.
- Choose one of the following from **Tier**:
 - Foundational IT core services**
 - Mission critical**
 - Critical**
 - Important**
 - Non critical**
- 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**.

- (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.
- (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*.
 - To create the policy, choose **Create**.

To create resiliency policies based on a suggested policy

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

The **Create resiliency policy** page displays.

- Enter a name for the resiliency policy.
- (Optional) Enter a description for the policy.
- 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**
- 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 AWS Resilience Hub resiliency assessments

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.

Running resiliency assessments

You can run a resiliency assessment report 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** table.

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

Review

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, manually added application, AppRegistry 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 Amazon 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"](#) 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.

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 alarms for your updated application.

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 AWS 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 AWS CloudFormation console.

To create an alarm based on AWS Resilience Hub recommendations, you must create an AWS 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"](#) 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 AWS 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 SOP 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 AWS CloudFormation template to store alarm details that is in-turn used for creating the alarms in Amazon CloudWatch. You can access the AWS CloudFormation template using Amazon S3 URL, and can download and place it into your code pipeline or create a stack through the AWS 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 AWS 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 AWS CloudFormation template from the AWS Resilience Hub recommendations is a good start. Use the AWS CloudFormation template to generate an AWS 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 AWS CloudFormation template for the recommended SOPs and include them in your code base.

Create an AWS 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"](#) 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 Amazon Fault Injection Service experiments

This section describes how to create and run Amazon 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. Majority of the AWS FIS experiments invoke Systems Manager automation actions to perform disruptions and monitor the alarms, and other AWS FIS experiments use only AWS FIS automation actions that are customised 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. AWS FIS experiments can be accessed from either AWS Resilience Hub ([the section called “Viewing fault injection experiments”](#)) or AWS FIS console ([AWS FIS](#)) .

Topics

- [Creating AWS FIS experiments from the operational recommendations](#)
- [Running an AWS FIS experiment from AWS Resilience Hub](#)
- [Viewing fault injection experiments](#)
- [Amazon Fault Injection Service experiment failures/status check](#)

Creating AWS FIS experiments from the operational recommendations

AWS Resilience Hub recommends that you test your application after you run an assessment report. You can access and run these experiments from your application's Assessment report.

AWS Resilience Hub provides a list of AWS FIS experiments, which are Systems Manager documents with testing parameters. When you select an AWS FIS experiment from the list, AWS Resilience Hub creates an AWS CloudFormation template with the parameters you define in the Systems Manager document. After the creation of the AWS CloudFormation stack, you can see your provisioned AWS FIS experiments for your application.

The AWS CloudFormation template consists of an IAM role for each Systems Manager document, with the minimum permissions required to run.

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

To create an AWS CloudFormation template for the AWS FIS experiment

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"](#) and then return to this step.
6. Under **Operational recommendations**, choose **Fault injection experiments**.
7. Select all the tests you want to include.
8. Choose **Create CloudFormation template**. This can take up to a few minutes to create the AWS CloudFormation template.

9. Choose **Templates**.

You can view the newly created AWS CloudFormation template in **Templates** table.

Complete the following procedure to include the 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 AWS FIS experiment template you just created.

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

- **Test name** – Name of the test that you have created for your application.
- **Description** – Describes the objective of the test.
- **State** – Indicates the current implementation state of the test.

This column displays one of the following values:

- **Implemented** – Indicates that the test is implemented in your application.
 - **Not implemented** – Indicates that the test is not implemented or included in your application.
 - **Excluded** – Indicates that the test is excluded from the application.
 - **Inactive** – Indicates that the test is deployed to AWS FIS, but it has not run in the last 30 days.
 - **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 experiment 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 test 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.

Running an AWS FIS experiment from AWS Resilience Hub

In your application, you must first create an AWS FIS experiment template from the operational recommendations before AWS Resilience Hub can run the AWS FIS experiment.

To start an AWS FIS experiment

1. In the left navigation menu, choose **Applications**.
2. From **Applications** table, open an application.
3. Choose **Fault injection experiments** tab.
4. Select the radio button before the experiment template that was used to create the experiment you want to run from the **Experiment templates** table, and then choose **Start experiment**.

To stop an AWS FIS experiment

1. In the left navigation menu, choose **Applications**.
2. From **Applications** table, open an application.
3. Choose **Fault injection experiments** tab.
4. Select the radio button before the experiment from the **Experiment** table, and then choose **Stop experiment**.

Viewing fault injection 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 AWS FIS experiments from the dashboard, choose **Dashboard** from the left navigation menu. In **Experiments** table, you can identify the implemented AWS FIS experiments using the following information:

- **Experiment ID** – Identifier of the AWS FIS experiment.

- **Experiment template ID** – Identifier of the AWS FIS experiment template that was used to create the AWS FIS experiment.
- **Source template** – Provides the Amazon Resource Name (ARN) of the AWS CloudFormation stack that contains details of the AWS FIS experiment.
- **State** – Indicates if the AWS FIS experiment is successfully completed or not.

To view the implemented AWS FIS experiments from applications

1. In the left navigation menu, choose **Applications**.
2. From **Applications** table, open an application.
3. Choose **Fault injection experiments**.
4. Choose **Experiment** tab.

In the **Experiment** tab, you can see a list of active AWS FIS experiment in the **Experiment** table.

In **Experiments** table, you can identify the implemented AWS FIS experiment using the following information:

- **Test name** – Name of the AWS Resilience Hub recommended test that was used to create the AWS FIS experiment.
- **Experiment ID** – Identifier of the AWS FIS experiment.
- **Description** – Describes the objective of the AWS FIS experiment.
- **Creation time** – Date and time when the AWS FIS experiment was created.
- **Last update time** – Date and time when the AWS FIS experiment was last updated.
- **Source template** – Provides the Amazon Resource Name (ARN) of the AWS CloudFormation stack that contains details of the AWS FIS experiment.

To view the recommended 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 **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 **Fault injection experiments** tab.

In **Fault injection experimentation templates** table, you can understand more about the recommended tests using the following information:

- **Name** – Name of the recommended test.
- **Description** – Describes the objective of the test.
- **State** – Indicates the current implementation state of the test.

This column displays one of the following values:

- **Implemented** – Indicates that the test is implemented in your application.
- **Not implemented** – Indicates that the test is not implemented or included in your application.
- **Excluded** – Indicates that the test is excluded from the application.
- **Inactive** – Indicates that the test is deployed to AWS FIS, but it has not run in the last 30 days.
- **Configuration** – Indicates if there are any pending configuration dependencies that needs to be addressed.

- **Type** – Indicates the type of test.
- **AppComponent** – Indicates the Application Components (AppComponents) that are associated with this test. For more information about supported AppComponent, see [Grouping resources in an AppComponent](#).
- **Risk** – Indicates the risk level of the test failure. Risk levels are indicated using **High**, **Medium**, and **Low** to indicate high, moderate, and low risk levels, respectively.
- **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.

Amazon 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 experiments from assessments** procedure in [the section called “Viewing fault injection experiments”](#).

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 FISaws : 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 AWS 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 explain 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.	$T = ((\text{Total number of tests implemented}) + (\text{Total number of tests excluded})) / (\text{Total number of tests recommended})$ <p>Parts of the formula are as follows:</p>	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
	<p>Note</p> <p>To calculate the resiliency score, the recommended tests must have run successfully in the last 30 days for AWS Resilience Hub to consider it as implemented.</p>	<ul style="list-style-type: none"> • Total number of tests configured – Indicates the total number of tests configured when the AWS CloudFormation template is created and uploaded in the AWS CloudFormation 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 recommended tests you have excluded from the application. 	<p>is calculated as follows:</p> $T = (10 + 5) / 20$ <p>That is, $T = .75$ or 75 points</p>

Scoring component	Description	Formula	Example
Alarms coverage (A)	<p>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.</p> <div data-bbox="367 827 760 1331" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>To calculate the resiliency score, the recommended alarms should be in Ready state for AWS Resilience Hub to consider it as implemented.</p> </div>	<p>$A = ((\text{Total number of alarms implemented}) + (\text{Total number of alarms excluded})) / (\text{Total number of alarms recommended})$</p> <p>Parts of the formula are as follows:</p> <ul style="list-style-type: none"> • 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 	<p>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:</p> $A = (10 + 5) / 20$ <p>That is, A = .75 or 75 points</p>

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.	$S = ((\text{Total number of SOPs implemented}) + (\text{Total number of SOPs excluded})) / (\text{Total number of SOPs recommended})$ <p>Parts of the formula are as follows:</p> <ul style="list-style-type: none"> • 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. 	<p>If you have implemented 10 and excluded 5 SOPs out of 20 AWS Resilience Hub recommended SOPs, the SOP coverage is calculated as follows:</p> $S = (10 + 5) / 20$ <p>That is, $S = .75$ or 75 points</p>

Scoring component	Description	Formula	Example
RTO/RPO compliance (P)	A normalized score (0 -100 points) based on the application meeting its resiliency policy.	$P = \frac{\text{Total weights of disruption types meeting the application's resiliency policy}}{\text{Total weights of all disruption types}}$	<p>If you application resiliency policy meets only for Availability Zone (AZ) and Infrastructure disruption types, the resiliency policy score (P) is calculated as follows:</p> <ul style="list-style-type: none"> If you have set regional RTO and RPO targets, P is calculated as follows: $P = \frac{(20 + 30)}{100}$ <p>That is, P = .5 or 50 points</p> If you have not set regional RTO and RPO targets, P is calculated as follows: $P = \frac{(22.22 + 33.33)}{99.9}$ <p>That is, P = .55 or 55 points</p>

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
Resiliency score for application (RS)	A normalized resiliency score (0 -100 points) based on your application meeting its resiliency 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 calculated using the following formula: $RS = (T * Weight(T) + A * Weight(A) + S * Weight(S) + P * Weight(P)) / (Weight(T) + Weight(A) + Weight(S) + Weight(P))$	<p>Formulas to calculate the coverage of each recommendation type table are as follows:</p> <ul style="list-style-type: none"> • Test coverage (T) = .75 • Alarms (A) = .75 • SOPs (S) = .75 • Meeting resiliency policy (P) = .5 <p>The resiliency score per application is calculated as follows:</p> $RS = ((.75 * .2) + (.75 * .2) + (.75 * .2) + (.5 * .4)) / (.2$

Scoring component	Description	Formula	Example
			$+ .2 + .2$ $+ .4)$ <p>That is, RS = .65 or 65 points</p>

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	<p>The resiliency score per AppComponent and per disruption type is calculated using the following formula:</p> $RSao = (T * Weight(T) + A * Weight(A) + S * Weight(S) + P * Weight(P)) / (Weight(T) + Weight(A) + Weight(S) + Weight(P))$	<p>RSao assumptions for all the recommendation types are as follows:</p> <ul style="list-style-type: none"> • Test coverage (T) = .75 • Alarms (A) = .75 • SOPs (S) = .75 • Meeting resiliency policy (P) = .5

Scoring component	Description	Formula	Example
	<p>average of all the recommendation types.</p> <p>That is: $RS_{ao} = \text{Weighted Average (T, A, S, P)}$</p> <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.</p>		<p>The resiliency score per AppComponent and disruption type is calculated as follows:</p> $RS_{ao} = ((.75 * .2) + (.75 * .2) + (.75 * .2) + (.5 * .4)) / (.2 + .2 + .2 + .4)$ <p>That is, $RS_{ao} = .65$ or 65 points</p>

Scoring component	Description	Formula	Example
Resiliency score per AppComponent (RSa)	<p>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)</p> <p>The values for T, A, S, P are calculated for all the recommended tests, alarms, SOPs, and meeting resiliency policy of the AppComponent.</p>	<p>The resiliency score per AppComponent is calculated using the following formula:</p> $RSa = (T * Weight(T) + A * Weight(A) + S * Weight(S) + P * Weight(P)) / (Weight(T) + Weight(A) + Weight(S) + Weight(P))$	<p>RSa assumptions for all the recommendation types are as follows:</p> <ul style="list-style-type: none"> • Test coverage (T) = .75 • Alarms (A) = .75 • SOPs (S) = .75 • Meeting resiliency policy (P) = .5 <p>The resiliency score per AppComponent is calculated as follows:</p> $RSa = ((.75 * .2) + (.75 * .2) + (.75 * .2) + (.5 * .4)) / (.2 + .2 + .2 + .4)$ <p>That is, RSa = .65 or 65 points</p>

Scoring component	Description	Formula	Example
Resiliency score per disruption type (RSo)	<p>A normalized score (0 -100 points) based on meeting its resiliency policy.</p> <p>Resiliency score per disruption type is the weighted average of all the recommendation types.</p> <p>That is: $RSo = \text{Weighted Average (T, A, S, P)}$</p> <p>The values for T, A, S, P are calculated for all the recommended tests, alarms, SOPs, and meeting resiliency policy of the disruption type.</p>	<p>The resiliency score per disruption type is calculated using the following formula:</p> $RSo = (T * \text{Weight}(T) + A * \text{Weight}(A) + S * \text{Weight}(S) + P * \text{Weight}(P)) / (\text{Weight}(T) + \text{Weight}(A) + \text{Weight}(S) + \text{Weight}(P))$	<p>RSo assumptions for all the recommendation types are as follows:</p> <ul style="list-style-type: none"> • Test coverage (T) = .75 • Alarms (A) = .75 • SOPs (S) = .75 • Meeting resiliency policy (P) = .5 <p>The resiliency score per disruption type is calculated as follows:</p> $RSo = ((.75 * .2) + (.75 * .2) + (.75 * .2) + (.5 * .4)) / (.2 + .2 + .2 + .4)$ <p>That is, $RSo = .65$ or 65 points</p>

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 AWS CloudFormation

After you choose **Create CloudFormation template** in the **Operational recommendations** page, AWS Resilience Hub creates an AWS CloudFormation template that describes the specific alarm, standard operating procedure (SOP), or AWS FIS experiment for your application. The AWS 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 AWS 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 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 AWS 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 AWS 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 AWS 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 AWS 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 AWS CloudFormation needs to replace the resource.

Note

You can choose more advanced methods for managing your application resources with AWS CloudFormation such as [nesting stacks](#) or [referring to resource outputs in a separate AWS 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”](#)

Step 1: 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 \  
Publish your application and assign resiliency policy
```



```
--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"  
  }  
}
```

Step 2: 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 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": "Alarm by AWS ResilienceHub that reports when 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": "Alarm by AWS ResilienceHub 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": "Alarm by AWS ResilienceHub 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": "Alarm by AWS ResilienceHub 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": "Alarm by AWS ResilienceHub 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": "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": [
    {
      "appName": "computeappcomponent-nrz",
      "recommendationStatus": "MetCanImprove",
      "configRecommendations": [
        {
          "cost": {
            "amount": 0.0,
            "currency": "USD",
            "frequency": "Monthly"
          },
          "appName": "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
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
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 ECS service with launch type EC2 and EFS
storage, deployed in multiple AZs. AWS Backup is used to backup EFS and copy snapshots
in-region.",
"suggestedChanges": [
    "Add Auto Scaling Groups and Capacity Providers in multiple
AZs",
    "Change desired count of the setup",
    "Remove 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 EFS from backup.
(Estimate is based on averages, real time restore may vary).",
                    "expectedRpoInSecs": 86400,
                    "expectedRpoDescription": "Recovery Point Objective for EFS
from backups, derived from backup frequency"
                }
            },
            "optimizationType": "BestAZRecovery",
            "description": "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 EFS from backup.
(Estimate is based on averages, real time restore may vary).",
        "expectedRpoInSecs": 86400,
        "expectedRpoDescription": "Recovery Point Objective for EFS
from backups, derived from backup frequency"
      }
    },
    "optimizationType": "BestAttainable",
    "description": "EFS with backups configured",
    "suggestedChanges": [
      "Add additional availability zone"
    ],
    "haArchitecture": "MultiSite",
    "referenceId": "efs:config:with_backups:2020-04-01"
  }
}

```

```
    ]
  }
]
}
```

Step 3: 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 the [Data Privacy](#)

[FAQ](#). For information about data protection in Europe, see the [AWS Shared Responsibility Model and GDPR](#) blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-2](#).

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with 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](#)
- [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, depending on the work that you do in AWS Resilience Hub.

Service user – If you use the AWS Resilience Hub service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more AWS Resilience Hub features to do your work, you might need additional permissions. Understanding how access is

managed can help you request the right permissions from your administrator. If you cannot access a feature in AWS Resilience Hub, see [Troubleshooting AWS Resilience Hub identity and access](#).

Service administrator – If you're in charge of AWS Resilience Hub resources at your company, you probably have full access to AWS Resilience Hub. It's your job to determine which AWS Resilience Hub features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with AWS Resilience Hub, see [How AWS Resilience Hub works with IAM](#).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to AWS Resilience Hub. To view example AWS Resilience Hub identity-based policies that you can use in IAM, 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* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see [How to sign in to your AWS account](#) in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see [Signing AWS API requests](#) in the *IAM User Guide*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see [Multi-factor authentication](#) in the

AWS IAM Identity Center User Guide and [Using multi-factor authentication \(MFA\) in AWS](#) in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see [Tasks that require root user credentials](#) in the *IAM User Guide*.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see [What is IAM Identity Center?](#) in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An [IAM user](#) is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see [Rotate access keys regularly for use cases that require long-term credentials](#) in the *IAM User Guide*.

An [IAM group](#) is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier

to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see [When to create an IAM user \(instead of a role\)](#) in the *IAM User Guide*.

IAM roles

An [IAM role](#) is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by [switching roles](#). You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see [Using IAM roles](#) in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see [Creating a role for a third-party Identity Provider](#) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permission sets, see [Permission sets](#) in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see [Cross account resource access in IAM](#) in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.

- **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see [Forward access sessions](#).
- **Service role** – 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 [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.
- **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using an IAM role to grant permissions to applications running on Amazon EC2 instances](#) in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see [When to create an IAM role \(instead of a user\)](#) 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 is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see [Overview of JSON policies](#) in the *IAM User Guide*.

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**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

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 [Creating IAM policies](#) in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see [Choosing 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 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.

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.

Access control lists (ACLs)

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.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see [Access control list \(ACL\) overview](#) in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see [Permissions boundaries for IAM entities](#) in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see [How SCPs work](#) in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. 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 [Creating IAM 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. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. 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. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant

the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. 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. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

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. Statements must include either a Resource or a NotResource element. As a best practice, specify a resource using its [Amazon Resource Name \(ARN\)](#). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, 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 (or Condition *block*) lets you specify conditions in which a statement is in effect. The Condition element is optional. 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.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical AND operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical OR operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see [IAM policy elements: variables and tags](#) in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. 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. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

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 [What is ABAC?](#) 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
--------------------------------	-----

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see [AWS services that work with IAM](#) in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see [Switching to a role \(console\)](#) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see [Temporary security credentials in IAM](#).

Forward access sessions for AWS Resilience Hub

Supports forward access sessions (FAS)	Yes
--	-----

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. 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 [Creating 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. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see [Creating IAM policies](#) 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 AWS 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 [IAM Access Analyzer policy validation](#) 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 [Configuring MFA-protected API access](#) 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.

```
{
  "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:ListGroupsWithUser",
        "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.

```
{
  "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.

```
{
  "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.

```
{
  "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.

```
{
  "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.

```
{
  "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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
        "resiliencehub:UpdateApp"
      ],
      "Resource": [
        "arn:aws:resiliencehub:*:*:app/appId"
      ],
      "Condition": {
        "StringLike": { "resiliencehub:policyArn": "arn:aws:resiliencehub:us-west-2:111122223333: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 `AwsResilienceHubAssessmentPolicy` 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 "Step 6: 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.

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.

```
{
  "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 `AWSResilienceHubAssessmentExecutionPolicy` 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 `AWSResilienceHubAssessmentExecutionPolicy` managed policy. For more information about this managed policy, see [the section called "AWSResilienceHubAssessmentExecutionPolicy"](#).

Sample trust-policy.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

```
{
  "Role": {
    "Path": "/",
    "RoleName": "AWSResilienceHubAssessmentRole",
    "RoleId": "AROAQFOXMPL6TZ6ITKWND",
    "Arn": "arn:aws:iam::123456789012:role/AWSResilienceHubAssessmentRole",
    "CreateDate": "2020-01-17T23:19:12Z",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [{
        "Effect": "Allow",
        "Principal": {
          "Service": "resiliencehub.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }]
    }
  }
}
```

```
}  
}
```

Sample attach-policy-to-role

```
aws iam attach-role-policy --role-name AWSResilienceHubAssessmentRole --  
policy-arn arn:aws:iam::aws:policy/  
AWSResilienceHubAssessmentExecutionPolicy
```

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.

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

4. In the **Permissions policies** section of **Add permissions** page, enter `AWSResilienceHubAssessmentExecutionPolicy` 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 `AWSResilienceHubAssessmentExecutionPolicy` managed policy. For more information about this managed policy, see [the section called "AWSResilienceHubAssessmentExecutionPolicy"](#).

Sample trust-policy.json

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

Sample create-role

```
aws iam create-role --role-name AWSResilienceHubAssessmentRole --assume-role-policy-document file://trust-policy.json
```

Sample Output

```
{
  "Role": {
    "Path": "/",
    "RoleName": "AWSResilienceHubAssessmentRole2",
    "RoleId": "AROAT2GICMEDJML6EVQRG",
    "Arn": "arn:aws:iam::262412591366:role/AWSResilienceHubAssessmentRole2",
    "CreateDate": "2023-08-02T07:49:23+00:00",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
```

```
        "AWS": [
            "arn:aws:iam::262412591366:role/
AWSResilienceHubAssessmentRole"
        ],
        "Action": "sts:AssumeRole"
    }
}
```

Sample attach-policy-to-role

```
aws iam attach-role-policy --role-name AWSResilienceHubAssessmentRole --
policy-arn arn:aws:iam::aws:policy/
AWSResilienceHubAssessmentExecutionPolicy.
```

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 `AWSResilienceHubAssessmentExecutionPolicy` 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 `AWSResilienceHubAssessmentExecutionPolicy` 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "sts:AssumeRole"
      ],
      "Resource": "arn:aws:iam::primary_account_id:role/
AwsResilienceHubAdminAccountRole"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::primary_account_id:role/
AwsResilienceHubAssessmentEKSAccessRole"
      ]
    }
  ]
}
```

Trust policy for the scheduled assessment's role (`AwsResilienceHubPeriodicAssessmentRole`)

```
{
  "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:

```
{
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Resource": [
          "arn:aws:iam::secondary_account_id_1:role/AwsResilienceHubExecutorAccountRole",
          "arn:aws:iam::secondary_account_id_2:role/AwsResilienceHubExecutorAccountRole",
          ...
        ],
        "Action": [
          "sts:AssumeRole"
        ]
      }
    ]
  }
}
```

The trust policy for the admin role (`AwsResilienceHubAdminAccountRole`) is as follows:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::primary_account_id:role/caller_IAM_role"
      }
    }
  ]
}
```

```

    },
    "Action": "sts:AssumeRole"
  },
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::primary_account_id:role/
AwsResilienceHubPeriodicAssessmentRole"
    },
    "Action": "sts:AssumeRole"
  }
]
}

```

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 `AWSResilienceHubAssessmentExecutionPolicy` managed policy to the role and attach the executor role policy.

The executor role trust policy is as follows:

```

{
  {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "AWS": "arn:aws:iam::primary_account_id:role/AwsResilienceHubAdminAccountRole"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }
}

```

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*.

AWSResilienceHubAssessmentExecutionPolicy

You can attach the `AWSResilienceHubAssessmentExecutionPolicy` 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 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 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.
- Amazon 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 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.
- AWS Directory Service – Lists and provides Describe permissions for AWS 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:DescribeScalableTargets",
        "autoscaling:DescribeAutoScalingGroups",
        "backup:DescribeBackupVault",
        "backup:GetBackupPlan",
        "backup:GetBackupSelection",
        "backup:ListBackupPlans",
        "backup:ListBackupSelections",
        "cloudformation:DescribeStacks",
        "cloudformation:ListStackResources",
        "cloudformation:ValidateTemplate",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:GetMetricData",
        "cloudwatch:GetMetricStatistics",
        "datasync:DescribeTask",
        "datasync:ListLocations",
        "datasync:ListTasks",
        "devops-guru:ListMonitoredResources",
        "dlm:GetLifecyclePolicies",
        "dlm:GetLifecyclePolicy",
        "drs:DescribeJobs",
        "drs:DescribeSourceServers",
        "drs:GetReplicationConfiguration",
        "ds:DescribeDirectories",
        "dynamodb:DescribeContinuousBackups",
        "dynamodb:DescribeGlobalTable",
        "dynamodb:DescribeLimits",
        "dynamodb:DescribeTable",
```

```
"dynamodb:ListGlobalTables",
"dynamodb:ListTagsOfResource",
"ec2:DescribeAvailabilityZones",
"ec2:DescribeFastSnapshotRestores",
"ec2:DescribeFleets",
"ec2:DescribeHosts",
"ec2:DescribeInstances",
"ec2:DescribeNatGateways",
"ec2:DescribePlacementGroups",
"ec2:DescribeRegions",
"ec2:DescribeSnapshots",
"ec2:DescribeSubnets",
"ec2:DescribeTags",
"ec2:DescribeVolumes",
"ec2:DescribeVpcEndpoints",
"ecr:DescribeRegistry",
"ecs:DescribeCapacityProviders",
"ecs:DescribeClusters",
"ecs:DescribeContainerInstances",
"ecs:DescribeServices",
"ecs:DescribeTaskDefinition",
"ecs:ListContainerInstances",
"ecs:ListServices",
"eks:DescribeCluster",
"eks:DescribeFargateProfile",
"eks:DescribeNodegroup",
"eks:ListFargateProfiles",
"eks:ListNodegroups",
"elasticache:DescribeCacheClusters",
"elasticache:DescribeGlobalReplicationGroups",
"elasticache:DescribeReplicationGroups",
"elasticache:DescribeSnapshots",
"elasticfilesystem:DescribeFileSystems",
"elasticfilesystem:DescribeLifecycleConfiguration",
"elasticfilesystem:DescribeMountTargets",
"elasticfilesystem:DescribeReplicationConfigurations",
"elasticloadbalancing:DescribeLoadBalancers",
"elasticloadbalancing:DescribeTargetGroups",
"elasticloadbalancing:DescribeTargetHealth",
"fis:GetExperimentTemplate",
"fis:ListExperimentTemplates",
"fis:ListExperiments",
"fsx:DescribeFileSystems",
"lambda:GetFunctionConcurrency",
```

```
"lambda:GetFunctionConfiguration",
"lambda:ListAliases",
"lambda:ListVersionsByFunction",
"rds:DescribeDBClusterSnapshots",
"rds:DescribeDBClusters",
"rds:DescribeDBInstanceAutomatedBackups",
"rds:DescribeDBInstances",
"rds:DescribeDBProxies",
"rds:DescribeDBProxyTargets",
"rds:DescribeDBSnapshots",
"rds:DescribeGlobalClusters",
"resource-groups:GetGroup",
"resource-groups:ListGroupResources",
"route53-recovery-control-config:ListClusters",
"route53-recovery-control-config:ListControlPanels",
"route53-recovery-control-config:ListRoutingControls",
"route53-recovery-readiness:GetReadinessCheckStatus",
"route53-recovery-readiness:GetResourceSet",
"route53-recovery-readiness:ListReadinessChecks",
"route53:GetHealthCheck",
"route53:ListHealthChecks",
"route53:ListHostedZones",
"route53:ListResourceRecordSets",
"route53resolver:ListResolverEndpoints",
"route53resolver:ListResolverEndpointIpAddresses",
"s3:GetBucketLocation",
"s3:GetBucketLogging",
"s3:GetBucketObjectLockConfiguration",
"s3:GetBucketPolicyStatus",
"s3:GetBucketTagging",
"s3:GetBucketVersioning",
"s3:GetMultiRegionAccessPointRoutes",
"s3:GetReplicationConfiguration",
"s3:ListAllMyBuckets",
"s3:ListBucket",
"s3:ListMultiRegionAccessPoints",
"servicecatalog:GetApplication",
"servicecatalog:ListAssociatedResources",
"sns:GetSubscriptionAttributes",
"sns:GetTopicAttributes",
"sns:ListSubscriptionsByTopic",
"sqs:GetQueueAttributes",
"sqs:GetQueueUrl",
"ssm:DescribeAutomationExecutions",
```

```

    "states:DescribeStateMachine",
    "states:ListStateMachineVersions",
    "states:ListStateMachineAliases",
    "tag:GetResources"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "apigateway:GET"
  ],
  "Resource": [
    "arn:aws:apigateway:*::/apis/*",
    "arn:aws:apigateway:*::/restapis/*",
    "arn:aws:apigateway:*::/usageplans"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "s3:CreateBucket",
    "s3:PutObject",
    "s3:GetObject"
  ],
  "Resource": "arn:aws:s3:::aws-resilience-hub-artifacts-*"
},
{
  "Effect": "Allow",
  "Action": [
    "cloudwatch:PutMetricData"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "cloudwatch:namespace": "ResilienceHub"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "ssm:GetParametersByPath"
  ],

```

```

    "Resource": "arn:aws:ssm:*:*:parameter/ResilienceHub/*"
  }
]
}

```

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 – AWS Resilience Hub extends support for Amazon FSx for Windows File Server.	This AWS Resilience Hub policy allows you to read the Amazon FSx for Windows File Server configuration.	March 26, 2024
AWSResilienceHubAssessmentExecutionPolicy – AWS Resilience Hub extends support for AWS Step Functions.	This AWS Resilience Hub policy allows you to read the AWS Step Functions configuration.	October 30, 2023
AWSResilienceHubAssessmentExecutionPolicy – AWS Resilience Hub improves support for Amazon Relational Database Service (Amazon RDS).	This AWS Resilience Hub policy allows you to access resources on Amazon RDS while running assessments.	October 5, 2023
AWSResilienceHubAssessmentExecutionPolicy – New policy	This AWS Resilience Hub policy provides access to other AWS services for running assessments.	June 26, 2023

Change	Description	Date
AWS Resilience Hub started tracking changes	AWS Resilience Hub started tracking changes for its AWS managed policies.	June 15, 2023

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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<primary-account>:role/<invoker-role-or-current-iam-role>"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::<s3-bucket-name>/<path-to-state-file>"
    },
    {
```



```

    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::<primary-account>:role/<invoker-role-or-current-iam-
role>"
    },
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::<s3-bucket-name>"
  }
]
}

```

- 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.

```

{
  "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 `AWSResilienceHubAssessmentExecutionPolicy` 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<primary-account>:role/<invoker-role-or-current-iam-role>"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::<bucket-with-statefile-in-secondary-account>/<path-to-state-file>"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<primary-account>:role/<invoker-role-or-current-iam-role>"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::<bucket-with-statefile-in-secondary-account>"
    }
  ]
}
```

- 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<primary-account>:role/<invoker-role-or-current-iam-
role>"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::<bucket-with-statefile-in-secondary-account>/<path-
to-state-file>"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::<primary-account>:role/<invoker-role-or-current-iam-
role>"
      },
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::<bucket-with-statefile-in-secondary-account>"
    }
  ]
}
```

Note

If you are using the `AWSResilienceHubAssessmentExecutionPolicy` 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.

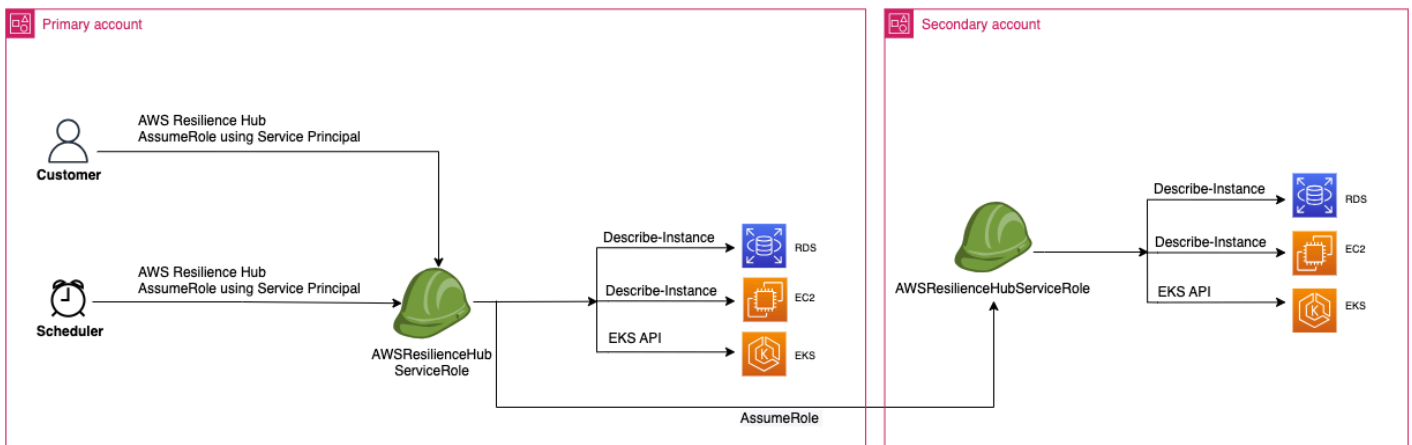
- For more information about Kubernetes role-based access control (RBAC) configuration, see [Using RBAC Authorization](#).

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:

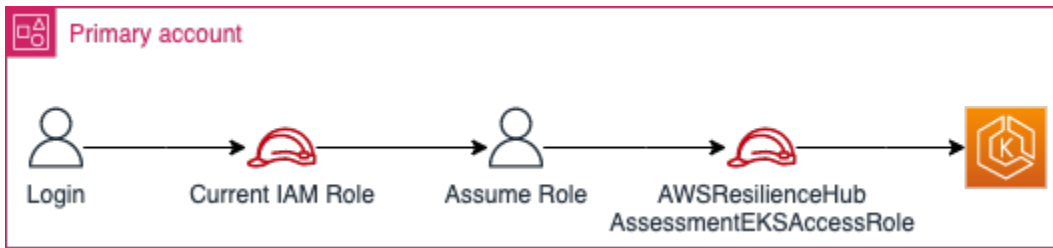
- If your application is configured to use role-based access for accessing resources, the invoker role or secondary account role passed to AWS Resilience Hub while creating an application will be used for accessing your Amazon EKS cluster during assessment.

The following conceptual diagram shows how AWS Resilience Hub accesses Amazon EKS clusters when the application is configured as a role-based application.

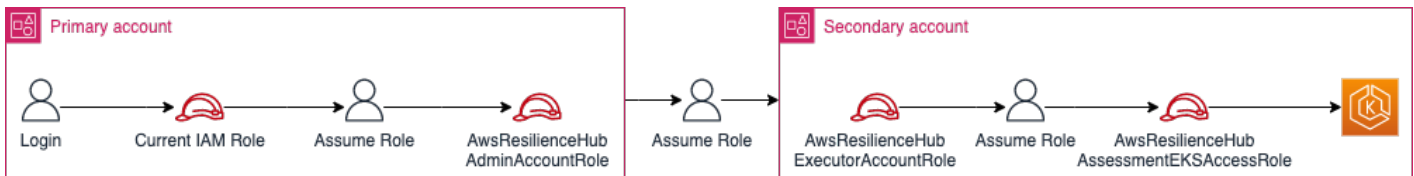


- If your application is configured to use the current IAM user for accessing resource, you must create a new IAM role with the name `AwsResilienceHubAssessmentEKSAccessRole` in the same account as that of the Amazon EKS cluster. This IAM role will then be used for accessing your Amazon EKS cluster.

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.

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

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
    "Principal": {
      "AWS": "arn:aws:iam::eks_cluster_account_id:role/
AwsResilienceHubExecutorRole"
    },
    "Action": "sts:AssumeRole"
  }
]
}
```

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
```

```

verbs:
  - get
  - list
- apiGroups:
  - karpenter.k8s.aws
resources:
  - awsnodetemplates
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
    - replicaset
    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
    verbs:
      - get
      - list
  - apiGroups:
      - karpenter.k8s.aws
    resources:
      - awsnodetemplates
    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:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowResilienceHubPublish",
      "Effect": "Allow",
      "Principal": {
        "Service": "resiliencehub.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:region:account-id:topic-name"
    }
  ]
}
```

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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowResilienceHubDecrypt",
      "Effect": "Allow",
      "Principal": {
        "Service": "resiliencehub.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:region:account-id:key/key-id"
    }
  ]
}
```

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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "resiliencehub:BatchUpdateRecommendationStatus",

```

```
    "Resource": "arn:aws:resiliencehub:us-west-2:12345678900:app/0e6237b7-23ba-4103-  
adb2-91811326b703"  
  }  
]  
}
```

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](#) whitepaper.

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.

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 AWS CloudFormation provisions and configures those resources for you.

When you use AWS 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 AWS CloudFormation templates

To provision and configure resources for AWS Resilience Hub and related services, you must understand [AWS CloudFormation templates](#). Templates are formatted text files in JSON or YAML. These templates describe the resources that you want to provision in your AWS CloudFormation stacks. If you're unfamiliar with JSON or YAML, you can use AWS CloudFormation Designer to help you get started with AWS CloudFormation templates. For more information, see [What is AWS CloudFormation Designer?](#) in the *AWS CloudFormation User Guide*.

AWS Resilience Hub supports creating `AWS::ResilienceHub::ResiliencyPolicy` and `AWS::ResilienceHub::App` in AWS 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 AWS 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 AWS CloudFormation

For more information about AWS CloudFormation, see the following resources:

- [AWS CloudFormation](#)
- [AWS CloudFormation User Guide](#)
- [AWS 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 AWS 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 AWS 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 resiliency 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”](#)
- [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”](#)
- [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 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: May 08, 2024**

Change	Description	Date
AWS Resilience Hub expands application resilience drift-detection capabilities	<p>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.</p> <p>For more information, see the following topics:</p> <ul style="list-style-type: none">• the section called “Drift detection”	May 8, 2024

- [the section called “Step 5: Setup scheduled assessment and drift notification”](#)

[AWS Trusted Advisor enhancements](#)

AWS Resilience Hub has expanded support for AWS Trusted Advisor by adding a check to identify unrecoverable Application Components (AppComponents).

March 28, 2024

For more information, see [the section called “AWS Trusted Advisor”](#).

[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.

March 26, 2024

For more information, see [the section called “Managing alarms”](#).

[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 application's resiliency. For applications 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 file system dependency on Microsoft Active Directory, for both in-Region and cross-Region deployments.

March 26, 2024

For more information, see the following topics:

- [the section called "Supported AWS Resilience Hub resources"](#)
- [the section called "AWSResilienceHubAssessmentExecutionPolicy"](#)
- [the section called "Grouping resources in an AppComponent"](#)

[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.

November 9, 2023

For more information, see [the section called “Understanding resiliency scores”](#).

[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.

November 9, 2023

For more information, see [the section called “Managing Amazon Fault Injection Service experiments”](#).

[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 information indicates the maximum possible estimated workload RTO and estimated workload RPO of your application from the latest successful 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 application's resiliency. AWS Resilience Hub analyzes the AWS Step Functions configuration including the state machine type (either Standard or Express workflows). In addition, AWS Resilience 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 permissions, see [the section called "AWSResilienceHubAssessmentExecutionPolicy"](#).

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 Amazon 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 application that was assessed. Using the exclude recommendations 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 beneficial if you are using a platform outside of one suggested, or have already implemented the recommendation in alternative method.

August 9, 2023

For more information, see the following topics:

- [the section called “Including or excluding operational recommendations”](#)
- [the section called “Limiting permissions to include or](#)

[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.

August 2, 2023

For more information about the AWS managed policy, see [the section called “AWSResilienceHubAssessmentExecutionPolicy”](#).

[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.

August 2, 2023

For more information, see the following topics:

- [the section called "Step 5: Setup scheduled assessment and drift notification"](#)
- [the section called "Editing application resources"](#)

[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](#)

June 27, 2023

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-scaling groups with Amazon Route 53 Application Recovery Controller (Route 53 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 Resilience Hub access permissions reference”](#).

[AWS managed policy updates](#)

Added a new policy that provides access to other AWS services for executing assessments.

June 26, 2023

For more information, see [the section called “AWSResilienceHubAssessmentsExecutionPolicy”](#).

[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.

May 2, 2023

For more information, see [the section called “AWS Resilience Hub access permissions reference”](#).

[AWS Trusted Advisor enhancements](#)

May 2, 2023

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"](#).

[Additional support for Amazon Simple Storage Service](#)

March 21, 2023

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](#)

[Additional support for Amazon Elastic Kubernetes Service](#)

March 21, 2023

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 applications, and receive assessments and recommendations for improving resiliency. Customers can add application resources using AWS CloudFormation, Terraform, AWS Resource Groups, and AppRegistry. Additionally, 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, ReplicationControllers, and Pods, AWS Resilience Hub will analyze the overall cluster resiliency. AWS Resilience Hub supports stateless Amazon EKS cluster workloads. The new capabilities are available in all the AWS Regions where AWS Resilience Hub is supported.

For more information, see the following topics:

- [the section called “Step 2: 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.

March 21, 2023

For more information, see the following topics:

- [the section called “Supported AWS Resilience Hub resources”](#)
- [What is Amazon Elastic File System?](#)

[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.

February 21, 2023

For more information, see [the section called “Editing application resources”](#).

[Support for application configuration parameters](#)

AWS Resilience Hub now 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.

February 21, 2023

For more information, see the following topics:

- [the section called “Application configuration parameters”](#)
- [the section called “Step 7: Configure the application configuration parameters”](#)
- [the section called “Updating application configuration parameters”](#)

[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 assess Amazon EBS snapshots by Amazon Data Lifecycle Manager and Amazon EBS fast snapshot restore (FSR).

February 21, 2023

For more information, see the following topics:

- [the section called “AWS Resilience Hub access permissions reference”](#)
- [Amazon Elastic Block Store \(Amazon EBS\)](#)

[Integration with AWS Trusted Advisor](#)

November 18, 2022

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”](#).

[Support for Amazon Simple Notification Service \(Amazon SNS\)](#)

AWS Resilience Hub now assesses applications using Amazon SNS by analyzing Amazon SNS configuration, including subscribers, and provides recommendations to meet the organization'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).

November 16, 2022

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 AppComponent"](#)

[Additional Support for Amazon Route 53 Application Recovery Controller \(Amazon Route 53 ARC\)](#)

AWS Resilience Hub now assesses Amazon Route 53 ARC for Elastic Load Balancing and Amazon Relational Database Service (Amazon RDS), which includes advising when Amazon Route 53 ARC would be beneficial. Extending AWS Resilience Hub, Amazon Route 53 ARC assessment support beyond AWS Auto Scaling Group (AWS ASG) and Amazon DynamoDB. Amazon Route 53 ARC provides high availability for your application, allowing you to quickly failover your entire application to a failover region.

November 16, 2022

For more information, see the following topics:

- [the section called “Supported AWS Resilience Hub resources”](#)
- [the section called “Identity and Access Management”](#)

[Additional Support for AWS Backup](#)

AWS Resilience Hub now assesses Amazon Route 53 ARC for Elastic Load Balancing and Amazon Relational Database Service (Amazon RDS), which includes advising when Amazon Route 53 ARC would be beneficial. Extending AWS Resilience Hub, Amazon Route 53 ARC assessment support beyond AWS Auto Scaling Group (AWS ASG) and Amazon DynamoDB. Amazon Route 53 ARC provides high availability for your application, allowing you to quickly failover your entire application to a failover region.

November 16, 2022

For more information, see the following topics:

- [the section called “Supported AWS Resilience Hub resources”](#)
- [the section called “Identity and Access Management”](#)

[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

AWS Glossary

For the latest AWS terminology, see the [AWS glossary](#) in the *AWS Glossary Reference*.