

Implementation Guide

Automations for AWS Firewall Manager



Automations for AWS Firewall Manager: Implementation 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

Solution overview	1
Architecture overview	3
Architecture diagram	3
Architecture details	6
AWS Lambda functions	6
AWS CloudFormation StackSets	7
AWS Firewall Manager integration	7
AWS Systems Manager Parameter Store	7
Amazon EventBridge	8
Amazon S3	8
Amazon DynamoDB	8
AWS services in this solution	9
Plan your deployment	11
Cost	11
Sample cost tables	11
Security	14
IAM roles	14
AWS Systems Manager Parameter Store	15
Supported AWS Regions	16
Quotas	16
Quotas for AWS services in this solution	16
AWS CloudFormation quotas	17
Deploy the solution	18
Prerequisites	18
Deployment process overview	18
AWS CloudFormation template	19
Step 1: (Optional) Install the prerequisite template	19
Step 1a. Launch the prerequisite stack	20
Step 1b. Manually activate AWS Firewall Manager (optional)	23
Step 2: Launch the stack	23
Step 3: Add and manage Firewall Manager policies	25
Access the Systems Manager Parameter Store history	25
Update the solution	27
Troubleshooting	28

AWS Config errors	28
Problem: Enabling AWS Config in the prerequisite stack doesn't work	28
Problem: Activating AWS Config using CloudFormation StackSets fails when creating the configuration recorder	29
Problem: AWS Config isn't activated in member accounts	29
Other errors	30
Problem: The FMS admin account-id isn't displayed in the Firewall Manager console	30
Problem: The CloudFormation StackSets instance displays as Outdated	30
Problem: InternalErrorException when creating a policy in Firewall Manager	31
Problem: Throttling exception with AWS APIs	32
Contact AWS Support	32
Create case	32
How can we help?	32
Additional information	33
Help us resolve your case faster	33
Solve now or contact us	33
Uninstall the solution	34
Using the AWS Management Console	34
Using AWS Command Line Interface	34
Deleting the Amazon S3 bucket	35
Use the solution	36
Set up the Systems Manager parameters	36
Create policies across OUs and Regions	36
Delete tags from policies	37
Delete Regional policies	38
Delete policies	38
Access compliance reports	38
Access CloudWatch Logs insights	39
Add CloudWatch Logs insights	40
Developer guide	42
Source code	42
List of policies and rule sets	42
Centralized WAF managed rules automation	42
Centralized security group audit checks	43
Centralized DDoS protection enablement	43
Centralized DNS Firewall rules automation	43

Policy manifest file	43
Customization guide	45
Change the default Firewall Manager security policy configuration	45
Apply different policies to different OUs and Regions	46
Example policy customization scenarios	48
Reference	51
Anonymized data collection	51
Other AWS WAF solution and resources	52
Contributors	52
Revisions	53
Notices	55

Centrally configure, manage, and audit firewall rules with Automations for AWS Firewall Manager

Publication date: *September 2020* (*last update: June 2024*)

The Automations for AWS Firewall Manager solution helps you centrally configure, manage, and audit firewall rules across your accounts and applications in [AWS Organizations](#). This solution uses [AWS Firewall Manager](#) to automatically deploy a set of managed rules for [AWS WAF](#) and audit checks for [Amazon Virtual Private Cloud](#) (Amazon VPC) security groups across your AWS accounts from a single place. This solution also provides [AWS Shield Advanced](#) customers with the option to deploy Distributed Denial of Service (DDoS) protection across accounts.

The process for defining policies and configuring rule sets in Firewall Manager can be challenging and time consuming. To help simplify this process, this solution deploys a set of AWS managed firewall rules and security group audit checks for you. Managed firewall rules provide a set of preconfigured rules to protect web applications running on [Amazon CloudFront](#), [Application Load Balancer](#), and [Amazon API Gateway](#). Security group audit checks continuously monitor and detect overly permissive security group rules to protect your Amazon VPC resources and improve your firewall posture.

This solution automates the onboarding process for Firewall Manager and sets up baseline rules and audit checks for AWS Organizations by allowing you to restrict policies for specific organizational units (OUs), Regions, or tagged resources within your AWS Organizations account. When you modify the installed [AWS Systems Manager Parameter Store](#) parameters, this solution updates and deploys the policies to the specified resources.

You can deploy the supplemental [AWS CloudFormation](#) supplemental template included in this solution into an AWS Organizations management account to configure the prerequisites for this solution automatically. For example:

- Checking that [all features](#) for AWS Organizations are activated.
- Designating an account as the admin account for Firewall Manager.
- Enabling [AWS Config](#) across an AWS Organization.

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

The intended audience for using this solution's features and capabilities in their environment includes solution architects, business decision makers, DevOps engineers, data scientists, and cloud professionals.

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

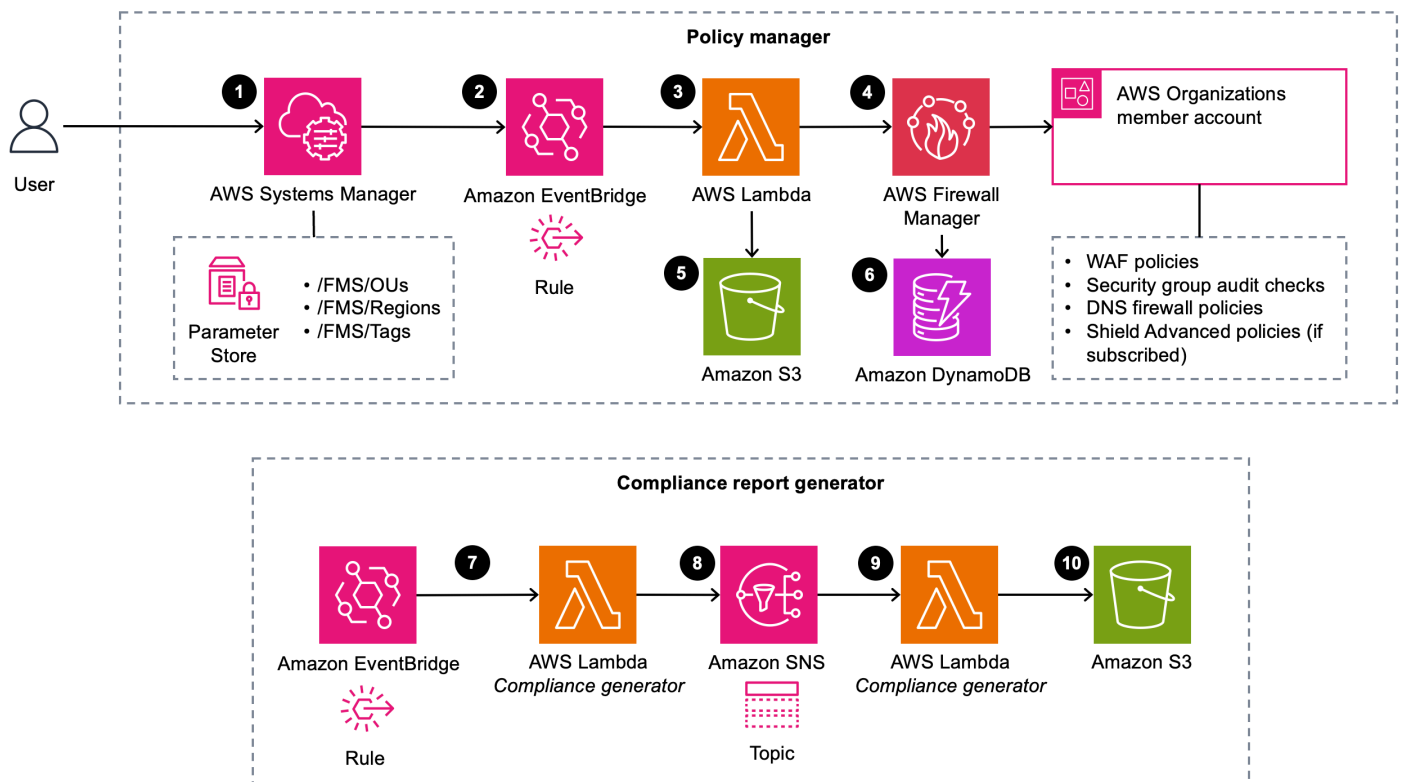
If you want to . . .	Read . . .
<p>Know the cost for running this solution.</p> <p>The estimated cost for running AWS resources for this solution in the US East (N. Virginia) Region is USD \$1,733.00 per month for a small organization or \$18,951.00 per month for a large organization.</p>	<p>Cost</p>
<p>Understand the security considerations for this solution.</p> <p>This solution uses Parameter Store to initiate create, read, update, and delete (CRUD) operations to the Firewall Manager policies.</p>	<p>Security</p>
<p>Know how to plan for quotas for this solution.</p>	<p>Quotas</p>
<p>Know which AWS Regions support this solution.</p>	<p>Supported AWS Regions</p>
<p>View or download the CloudFormation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this solution.</p>	<p>AWS CloudFormation template</p>
<p>Access the source code and optionally use the AWS Cloud Development Kit (AWS CDK) to deploy the solution.</p>	<p>GitHub repository</p>

Architecture overview

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

Architecture diagram

Deploying this solution with the default parameters deploys the following components in your AWS account.



Automations for AWS Firewall Manager solution architecture

Note

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

The architecture can be grouped into two separate workflows: policy manager and compliance report generator.

Policy manager

When the [CloudFormation](#) template deploys, an [AWS Systems Manager Parameter Store](#) containing three parameters is created, each with default values. The parameters that are created include **/FMS/OUs**, **/FMS/Regions**, and **/FMS/Tags**.

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

1. You can update these parameters using Systems Manager:
 - For the **/FMS/OUs** parameter, add organizational unit IDs to apply policies and rule sets to multiple OUs.
 - For the **/FMS/Regions** parameter, specify AWS Region names.
 - For the **/FMS/Tags** parameter, create *inclusion* and *exclusion* tags and add tags to specific resources within accounts to indicate resources for which policies and rule sets should be applied or not applied respectively. For information about setting up Parameter Store parameters, refer to [Scenarios for setting up the Systems Manager parameters](#).
2. An [Amazon EventBridge](#) rule uses an event pattern to capture the Systems Manager parameter update event.
3. An EventBridge rule invokes an [AWS Lambda](#) function.
4. The Lambda function installs a set of predefined Firewall Manager security policies across the user-specified OUs. The policies include an AWS WAF web access control list (ACL) consisting of AWS-managed rule sets and [Amazon VPC](#) security group audit policies. Additionally, if you have a subscription to [Shield Advanced](#), this solution deploys policies to protect against DDoS attacks.
5. The PolicyManager Lambda function fetches the policy manifest file from the [Amazon Simple Storage Service](#) (Amazon S3) bucket and uses the manifest file to create Firewall Manager security policies.
6. Lambda saves policies metadata in the [Amazon DynamoDB](#) table.

For a complete list of policies and rule sets that are installed, and information about the recommended policy default results and where they are contained, refer to [Scenarios for setting up the System Manager parameters](#).

Compliance report generator

When the CloudFormation stack deploys, it creates a time-based EventBridge rule, a Lambda function, an [Amazon Simple Notification Service](#) (Amazon SNS) topic, and an S3 bucket.

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

7. A time-based EventBridge rule invokes the `ComplianceGenerator` Lambda function.
8. The `ComplianceGenerator` Lambda function fetches Firewall Manager policies in each Region and publishes the list of policy IDs in the Amazon SNS topic.
9. The Amazon SNS topic invokes the `ComplianceGenerator` Lambda function with the payload `{PolicyId: string, Region: string}`.
10. The `ComplianceGenerator` Lambda function generates a compliance report for each of the policies and uploads the report in CSV format in an S3 bucket.

Architecture details

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

AWS Lambda functions

This solution uses Lambda functions to initiate prerequisite checks and the installation of policies and rule sets in OUs for Firewall Manager.

This solution uses the following Lambda functions:

- `PreReqManager` – This Lambda function checks and validates the following:
 - The prerequisite stack is deployed in the AWS Organizations primary account
 - The AWS Organizations all features option is activated
 - There is a delegated admin account assigned for Firewall Manager
 - Trusted access is activated between AWS Organizations and CloudFormation [StackSets](#)
 - AWS Config is activated across AWS Organizations for all member accounts

You can access log information for this Lambda function by following these instructions:

1. Sign in to the [Amazon CloudWatch console](#).
 2. Select **Logs** from the navigation menu, then **Log groups**.
 3. Select the log group named: `/aws/lambda/<Stack-Name>-xxx-PreReqManager-xxx`.
- `PolicyManager` – This Lambda function is responsible for managing Firewall Manager policies, such as creating, updating, and deleting the policies. The Lambda function fetches the policy manifest file from the S3 bucket and uses it to create Firewall Manager security policies. The manifest file can be modified at any time per requirement for policy configuration. The changes in the policy manifest are picked up with the next policy update event. The function saves policy metadata in the DynamoDB table.

You can access log information for this Lambda function by following these instructions:

1. Sign in to the [Amazon CloudWatch console](#).
2. Select **Logs** from the navigation menu, then **Log groups**.
3. Select the log group named: `/aws/lambda/<Stack-Name>-xxx-PolicyManager-xxx`.

- **ComplianceGenerator** – This Lambda function generates compliance reports for audit purposes. The reports are generated in CSV format and staged in an S3 bucket.
 1. Sign in to the [Amazon CloudWatch console](#).
 2. Select **Logs** from the navigation menu, then **Log groups**.
 3. Select the log group named: `/aws/lambda/<Stack-Name>-xxx-ComplianceGenerator-xxx`.

AWS CloudFormation StackSets

This solution uses service-managed CloudFormation StackSets with service-managed permissions to use AWS Config across the AWS Organization.

Note

The amount of time to turn on AWS Config depends on the number of member accounts and Regions under consideration. For example, in testing, it took approximately 90 minutes to turn on AWS Config across 6 accounts and 16 Regions for 2 OUs.

AWS Firewall Manager integration

This solution automatically installs policies and rule sets for Firewall Manager. By default, AWS WAF, security group, and [Amazon Route 53 Domain Name System \(DNS\) Firewall](#) security policies are installed. Additionally, if you have a subscription to Shield Advanced, Shield policies are also installed.

Firewall Manager policies are configured with auto-remediation activated for AWS WAF and Shield Advanced policies. If you want to customize policy deployment or another aspect of the solution, refer to the [README.md](#) file in the GitHub repository.

AWS Systems Manager Parameter Store

Parameter Store stores the solution's configuration parameters. You can use these parameters to specify OUs, Regions, and tags. The Parameter Store parameters allow you to easily extend policies and rule sets to multiple OUs and Regions. These parameters also allow you to specify inclusion and exclusion tags and apply these tags to specific resources in your accounts.

Additionally, administrators can view and modify the solution's parameters in one centralized location. You can add, edit, and remove parameter values to modify their selection across OUs, Regions, and tags. Corresponding Firewall Manager policies are updated automatically.

Amazon EventBridge

This solution uses the Amazon EventBridge rule to invoke Lambda functions when updates are made to Parameter Store for OUs, Regions, and tags. When the Lambda functions are initiated, policies and rule sets are installed in OUs and Regions (as updated by the user).

Amazon S3

The solution creates two S3 buckets in your account. One bucket stages the policy manifest file, and the other bucket is used by the ComplianceGenerator Lambda function to save compliance reports.

Amazon DynamoDB

This solution uses DynamoDB to save metadata created from Firewall Manager policies. The metadata is used to update and delete policies across specified OUs and Regions. The following is sample metadata from a Firewall Manager policy.

```
{
  "LastUpdatedAt": "2020-09-10T19:18:33.719Z",
  "PolicyId": "abcd1234-ab12-cd34-b99b-ab01cde2fg34",
  "PolicyName": "FMS-Shield-01",
  "PolicyUpdateToken": "1:AbCde1fGH2iJKLM34n05PQ==",
  "Region": "Global"
}
```

Important

Do not delete this table. It is used to perform create, update, and delete actions on the policies.

AWS services in this solution

AWS service	Description
AWS CloudFormation	Core. Deploys the AWS resources for this solution.
Amazon DynamoDB	Core. Stores metadata for this solution. The solution uses this metadata to perform create, update, and delete actions on policies.
AWS Firewall Manager	Core. Automatically deploys a set of managed rules for AWS WAF and audit checks for VPC security groups across your AWS accounts.
AWS Organizations	Core. Helps you centrally manage your accounts. This solution sets up baseline rules and audit checks for AWS Organizations.
Amazon S3	Core. Stores the policy manifest and compliance reports.
AWS Config	Supporting. Publishes events for resource changes. These events invoke Lambda functions to monitor compliance of the resources that Firewall Manager configures.
Amazon EventBridge	Supporting. Invokes Lambda functions for this solution when Parameter Store for OUs, Regions, and tags are updated.
AWS Lambda	Supporting. Initiates prerequisite checks and the installation of policies and rule sets in OUs for Firewall Manager.
Amazon SNS	Supporting. Invokes the ComplianceGenerator Lambda function.

AWS service	Description
AWS Systems Manager	Supporting. Stores the solution's configuration parameters.

Plan your deployment

This section describes the [cost](#), [security](#), [Regions](#), and other considerations prior to deploying the solution.

Cost

You are responsible for the cost of the AWS services used while running this solution. As of this revision, the cost to run the solution in the US East (N. Virginia) Region is approximately:

- **\$1,733.00 per month** for a small organization
- **\$18,951.00 per month** for a large organization

These costs are for the resources shown in the [Sample cost tables](#). The total cost to run this solution depends on the following:

- Number of policies installed
- Number of accounts managed
- Number of rule sets and web ACLs installed
- Number and invocation duration of Lambda functions
- Number of EventBridge events published

For example, for two CloudFront global policies and one Regional policy, the total policy cost is:

$$3 \text{ policies} \times \$100 = \$300 \text{ per month}$$

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

Sample cost tables

The following tables provide a sample cost breakdown for deploying this solution with the default parameters in the US East (N. Virginia) Region for one month.

Cost per month for a small organization

Assumptions:

- Accounts: 12 accounts across 2 OUs
- Number of AWS Regions: 3
- Subscription to AWS Shield Advanced: No
- Number of policies: 13
 - CloudFront global policy: AWS WAF global policy (\$100 x 1 global policy)
 - Regional policies:
 - AWS WAF Regional policy (\$100 x 3 Regions)
 - Security group content audit policy (\$100 x 3 Regions)
 - Security group usage audit policy (\$100 x 3 Regions)
 - DNS Firewall policy (\$100 x 3 Regions)

Note

The following cost estimate doesn't account for a subscription to AWS Shield Advanced. With the Shield Advanced subscription, the AWS WAF protection policy cost and the AWS WAF web ACL and rules cost are included. For additional information, refer to the [AWS Firewall Manager pricing](#) page.

Components	Quantity	Accounts	\$/month [USD]	Monthly Total [USD]
AWS Firewall Manager				
Policies	13	N/A	\$100.00	\$1,300.00
AWS WAF web ACL	4	12	\$5.00	\$240.00
AWS WAF rules	4 x 4	12	\$1.00	\$192.00

Components	Quantity	Accounts	\$/month [USD]	Monthly Total [USD]
Other AWS services*				
N/A	N/A	12	less than \$1.00	\$1.00
Total:				\$1,733.00

* Other AWS services include Lambda, EventBridge, CloudFormation StackSets, AWS Config, DNS Firewall, and Parameter Store.

Cost per month for a large organization

Assumptions:

- Accounts: 150 accounts across 20 OUs
- Number of AWS Regions: 10
- Subscription to AWS Shield Advanced: No
- Number of policies: 41
 - Global policy: AWS WAF global policy (\$100 x 1 global policy)
 - Regional policies:
 - AWS WAF Regional policy (\$100 x 10 AWS Regions)
 - Security group content audit policy (\$100 x 10 Regions)
 - Security group usage audit policy (\$100 x 10 Regions)
 - DNS Firewall policy (\$100 x 10 Regions)

Note

The following cost estimate doesn't account for a subscription to AWS Shield Advanced. With the Shield Advanced subscription, the AWS WAF protection policy cost and the AWS WAF web ACL and rules cost are included. For additional information, refer to the [AWS Firewall Manager pricing](#) page.

Components	Quantity	Accounts	\$/month [USD]	Monthly Total [USD]
AWS Firewall Manager				
Policies	41	N/A	\$100.00	\$4,100.00
AWS WAF web ACL	11	150	\$5.00	\$8,250.00
AWS WAF rules	4 × 11	150	\$1.00	\$6,600.00
Other AWS services*				
N/A	N/A	150	less than \$1.00	\$1.00
Total:				\$18,951.00

* Other AWS services include Lambda, EventBridge, CloudFormation StackSets, AWS Config, DNS Firewall, and Parameter Store.

Security

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

IAM roles

[AWS Identity and Access Management](#) (IAM) roles allow customers to assign granular access policies and permissions to services and users on the AWS Cloud. This solution creates IAM roles that grant the solution's Lambda functions access to create Regional resources.

Permissions required by the prerequisite stack

The appropriate IAM permissions are required to fulfill the prerequisites. These permissions include allowing trusted access for AWS services with AWS Organizations, creating and deleting stack set instances to configure AWS Config in member accounts, configuring the Firewall Manager admin, and recording Lambda events in [CloudWatch Logs](#).

Permissions required by the primary stack

The appropriate IAM permissions are required to manage Firewall Manager policies. These permissions include:

- Creating and deleting Firewall Manager policies for AWS WAF, Shield, VPC Security Groups, and DNS Firewall
- Reading and writing DynamoDB tables with policy metadata
- Reading Systems Manager parameter information
- Recording Lambda events in CloudWatch Logs.

Additionally, the `ComplianceGenerator` Lambda function needs permission to describe all Firewall Manager policies, generate compliance reports, and upload them in an S3 bucket.

AWS Systems Manager Parameter Store

This solution uses Parameter Store to initiate create, read, update, and delete (CRUD) operations to the Firewall Manager policies. Systems Manager parameters created by this solution must be secured. Access should only be granted to a specific principal or user. A user with malicious intent that has access to these parameters can cause undesirable Firewall Manager policy operations, such as deleting policies. Such operations may be initiated across several member accounts in AWS Organizations.

An IAM-user, role, or federated user is denied access by default. A user must be explicitly authorized to [perform an action](#). Unless a user receives explicit permission to access these Systems Manager parameters, changes cannot be made to the solution parameters. Additionally, you can use *explicit deny* to prevent further access to these resources as shown in the following example policy. This example policy can be assigned to users to prevent access to the DynamoDB table and Systems Manager parameters resources.

```
{
```

```
"Version": "2012-10-17",
"Statement": [{
  "Action": [
    "dynamodb:*"
  ],
  "Resource": "arn:aws:dynamodb:<region>:<account-id>:table/<table-name>",
  "Effect": "Deny",
  "Sid": "FMSDDBSecure"
},
{
  "Action": "ssm:*"
  "Resource": [
    "arn:aws:ssm:<region>:<account-id>:parameter/FMS/OU",
    "arn:aws:ssm:<region>:<account-id>:parameter/FMS/Regions",
    "arn:aws:ssm:<region>:<account-id>:parameter/FMS/Tags"
  ],
  "Effect": "Deny",
  "Sid": "FMSSMSecure"
}
]
```

Supported AWS Regions

Although AWS Organizations and Firewall Manager are available globally, both AWS services use the US East (N. Virginia) Region as their data plane. As a result, the service clients for these AWS services must be created with the us-east-1 endpoint. Deploying in another AWS Region will work, but if there are AWS Organizations service control policies or custom firewall rules restricting traffic from transmitting out of the Region, then these APIs will fail. If you have restrictions in place, then we recommend deploying the solution in the US East (N. Virginia) Region.

Quotas

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

Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the [services implemented in this solution](#). For more information, see [AWS service quotas](#).

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

AWS CloudFormation quotas

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

Deploy the solution

This solution uses [AWS CloudFormation templates and stacks](#) to automate its deployment. The CloudFormation template specifies the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources that are described in the template.

Prerequisites

If you don't have Firewall Manager configured in your AWS Organizations primary account, then you must deploy the solution's prerequisite template first. This template must be deployed in the AWS Organizations management account with the AWS Organizations all features option activated prior to deploying the template.

For more information, refer to [Step 1: \(Optional\) Install the prerequisite template](#).

Deployment process overview

Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Before you launch the solution, review the [cost](#), [architecture](#), [network security](#), and other considerations discussed earlier in this guide.

Time to deploy: Approximately three minutes

[Step 1: \(Optional\) Install the prerequisite template](#)

[Step 2. Launch the stack](#)

[Step 3: Add and manage Firewall Manager policies](#)

Important

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered through this survey. Data collection is subject to the [AWS Privacy Notice](#).

To opt out of this feature, download the template, modify the AWS CloudFormation mapping section, and then use the AWS CloudFormation console to upload your updated

template and deploy the solution. For more information, see the [Anonymized data collection](#) section of this guide.

AWS CloudFormation template

You can download the CloudFormation template for this solution before deploying it.

[View template](#)

aws-fms-automations.template - Use this template to launch the solution and all associated components. The default configuration deploys the core and supporting services found in the [AWS services in this solution](#) section, but you can customize the template to meet your specific needs.

Note

AWS CloudFormation resources are created from AWS CDK constructs.

This AWS CloudFormation template deploys the Automations for AWS Firewall Manager solution in the AWS Cloud.

Note

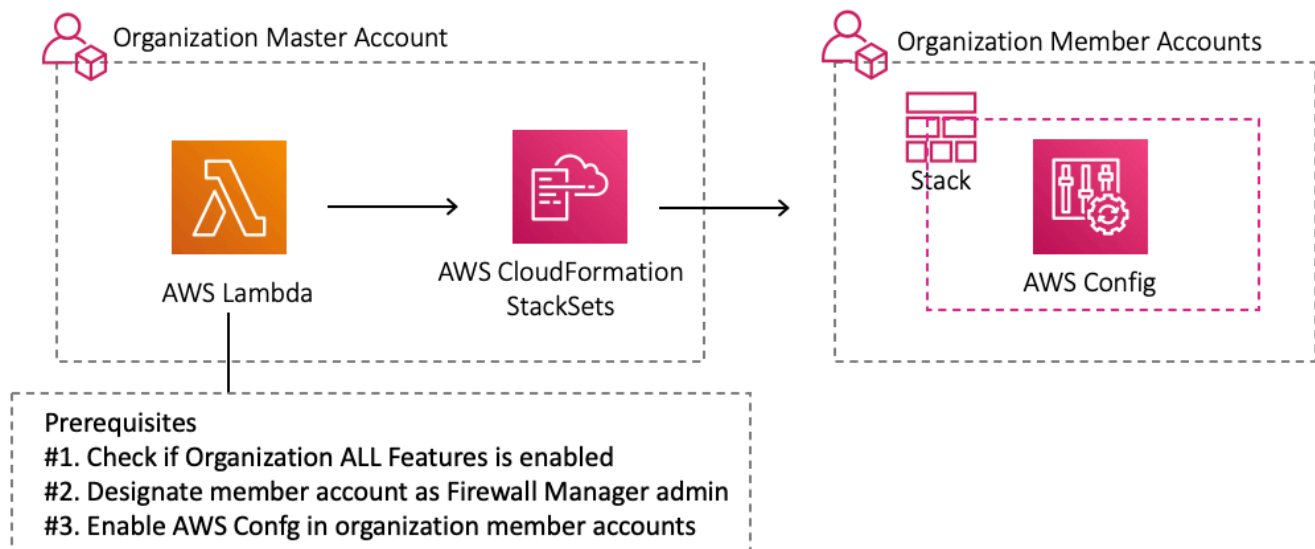
If you have previously deployed this solution, see [Update the solution](#) for update instructions.

Step 1: (Optional) Install the prerequisite template

Important

If Firewall Manager is already configured in your AWS Organizations management account, proceed to [Step 2: Launch the stack](#).

Installing the Firewall Manager prerequisite template in an AWS Organizations primary account with the default parameters builds the following environment in the AWS Cloud.



Architecture: Turn on prerequisites

When the template is deployed in an AWS Organizations primary account, a Lambda function checks for the following prerequisites:

1. The **AWS Organizations All Features** function is activated.
2. The AWS Firewall Manager admin is configured.
3. Optional: AWS Config is activated.

Note

This check is done when you activate AWS Config (set to Yes) during deployment of the prerequisite template. See [Step 1a: Launch the prerequisite stack](#) for more information.

The Lambda function installs the prerequisites. If there are errors during prerequisite installation, a stack rollback occurs with an error message.

Step 1a. Launch the prerequisite stack

This automated AWS CloudFormation template deploys the Firewall Manager prerequisite template in the AWS Cloud.

[View template](#)

aws-fms-prereq.template - Use this template to launch the solution prerequisite template. The default configuration deploys Lambda functions, CloudFormation StackSets, and AWS Config resources.

Note

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

1. Sign in to the [AWS Management Console](#) and select the button to launch the `aws-fms-prereq.template` CloudFormation template.

[Launch solution](#)

2. The template launches in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.

Note

Although AWS Organizations and Firewall Manager are available globally, both AWS services use the US East (N. Virginia) Region as their data plane. See [Supported AWS Regions](#) for more information.

3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, see [IAM and AWS STS quotas, name requirements, and character limits](#) in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
FMS Admin Account ID	<i><Requires input></i>	Add your Firewall Manager service admin account ID, if you have already configured your Firewall Manager admin account. Otherwise, specify an AWS Organizations member account ID that you want as designated Firewall Manager admin account.
Enable Config	Yes	Activate AWS Config across the organization for the resources required by Firewall Manager. If you already have AWS Config activated, select No.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review and create** page, review and confirm the settings. Select the box acknowledging that the template will create IAM resources.
9. Choose **Submit** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation Console in the **Status** column. You should receive a CREATE_COMPLETE status in approximately 10 minutes.

Note

When installing the prerequisite template, you have the option to designate a separate account in your organization as the Firewall Manager administrator account. If you select this option, you must manually install the `aws-fms-automations` template in the

designated account after installing the prerequisite template in your AWS Organizations management account.

Step 1b. Manually activate AWS Firewall Manager (optional)

Use the following procedure to activate AWS Firewall Manager in AWS Organizations.

1. Activate **AWS Organizations All Features**.
2. Activate **AWS Config** on all Organizations member accounts.
3. Designate a member account as **Firewall Manager Admin**.

For additional information to enable Firewall Manager, refer to [AWS Firewall Manager prerequisites](#) in the *AWS WAF, AWS Firewall Manager, and AWS Shield Advanced Developer Guide*.

Step 2: Launch the stack

Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately three minutes

1. Sign in to the [AWS Management Console](#) and select the button to launch the `aws-fms-automations.template` CloudFormation template.

[Launch solution](#)

2. The template launches in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.

Note

Although AWS Organizations and Firewall Manager are available globally, both AWS services use the US East (N. Virginia) Region as their data plane. See [Supported AWS Regions](#) for more information.

3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.

4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, see [IAM and AWS STS quotas, name requirements, and character limits](#) in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Compliance Reporting	Yes	Choose Yes or No based on your preference for generating compliance reports for your Firewall Manager security policies.

6. Select **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Select the boxes acknowledging that the template will create IAM resources and an auto-expand capability.
9. Choose **Create stack** to deploy the stack.

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

Note

In addition to the primary Lambda functions, this solution includes the `solution-helper` Lambda function, which runs only during initial configuration or when resources are updated or deleted.

When you run this solution, you will notice both Lambda functions in the AWS console. Only the primary functions are regularly active. However, you must not delete the `solution-helper` function, as it is necessary to manage associated resources.

Step 3: Add and manage Firewall Manager policies

You can add Firewall Manager policies across multiple OUs and Regions for your business needs. Using Systems Manager parameters, you can manage Regions and OUs where the policies get created or deleted, and you can manage the resources under scope using the **Tag** parameter. Use the following procedure to update each parameter:

1. Sign in to the [AWS Systems Manager console](#).
2. On navigation menu, under **Application Management**, select **Parameter Store**.
3. Select the parameter to update and choose **Edit**.
4. Update the value.
5. Choose **Save changes**.

You can update these parameters at any time and as many times as needed to meet your use cases and preferences for setting up your OUs, Regions, and tags. These parameters have the following format:

- /FMS/<PolicyID>/OUs: <StringList>
- /FMS/<PolicyID>/Regions: <StringList>
- /FMS/<PolicyID>/Tags: <String>

For examples on updating these parameters, refer to [Scenarios for setting up the Systems Manager parameters](#).

Access the Systems Manager Parameter Store history

Use the following steps to identify the person that invoked a change to the parameters in Parameter Store:

1. Sign in to the [AWS Systems Manager console](#).
2. On the navigation menu, under **Application Management**, select **Parameter Store**.
3. Select the parameter and choose **View Details**.
4. Choose **History**.

Note

If you want to customize the default policies or want different policies being applied to different OUs and Regions, refer to the [Customization guide](#). This section describes how you can use `aws-fms-policy.template` to apply a different set of policies to different OUs or Regions.

Update the solution

If you have previously deployed the solution, follow this procedure to update the solution's CloudFormation stack to get the latest version of the solution's framework.

Note

This solution supersedes the AWS Centralized WAF and VPC Security Group Management solution. If you previously deployed the solution, follow this procedure to safely migrate to the latest version of Automations for AWS Firewall Manager.

1. Follow the instructions in [Step 2: Launch the stack](#) and [Step 3: Add and manage Firewall Manager policies](#).
2. Additionally, you can configure the policies to meet custom requirements by changing values in the policy manifest file. For more information, refer to the [Customization guide](#).
3. Confirm that your new Firewall Manager policies are consistent with your requirements.
4. Delete the previously deployed version of the solution by following these instructions:
 - a. Sign in to the [AWS CloudFormation console](#).
 - b. Select the existing `aws-centralized-waf-and-security-group-management` CloudFormation stack.
 - c. Choose **Delete**.

You have now safely migrated to the latest version of this solution and the supported Firewall Manager policies.

Troubleshooting

This section provides troubleshooting instructions when deploying the solution.

Before addressing the following common errors, you can adjust the level of detail in the CloudWatch Logs. For more details, refer to [Amazon CloudWatch logs insights](#).


The [the section called “AWS Config errors”](#) and [the section called “Other errors”](#) resolution sections provide instructions to mitigate known errors. If these instructions don't address your issue, [the section called “Contact AWS Support”](#) provides instructions for opening an AWS Support case for this solution.

AWS Config errors

This section addresses known errors with AWS Config when deploying or using this solution.

Problem: Enabling AWS Config in the prerequisite stack doesn't work

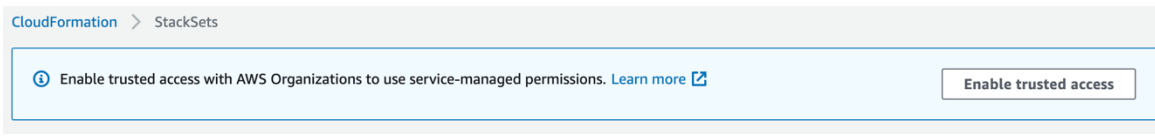
The following error occurs when you deploy solution's `aws-fms-prereq.template` CloudFormation template with the **Enable Config** parameter set to Yes.

PreReqManagerCR	 CREATE_FAILED	Received response status [FAILED] from custom resource. Message returned: stack set instance creation failed logs: /aws/lambda/tc04-3-PreReqManagerFunction80D2ED4C-X8xz06V4RXIx at sendResponse (/var/task/index.js:155:15) at Runtime.exports.handler (/var/task/index.js:132:18) at processTicksAndRejections (internal/process/task_queues.js:95:5) (RequestId: 3ed6460e-2b8f-4dcf-87cb-8476cda9cb2f)
-----------------	--	---

Reason: Trusted access for CloudFormation StackSets can **only** be enabled using the AWS CloudFormation console. Refer to [Enabling trusted access with AWS CloudFormation Stacksets](#) in the *AWS Organizations User Guide*.

Resolution

1. Sign in to the [AWS CloudFormation console](#).
2. From the navigation menu, choose **StackSets**.
3. Choose **Activate trusted access**. Providing a registered delegated administrator is optional.



4. Deploy the `aws-fms-prereq.template` again.

Problem: Activating AWS Config using CloudFormation StackSets fails when creating the configuration recorder

The following error occurs in the StackSets console:

```
ResourceLogicalId:ConfigRecorder, ResourceType:AWS::Config::ConfigurationRecorder, ResourceStatusReason:Failed to put configuration recorder 'StackSet-FMS-EnableConfig-CloudFront-2765adb1-71a9-4a3e-9bbb-535c4efdf35e-ConfigRecorder-1V0GK1MU9SVGJ' because maximum number of configuration recorders: 1 is reached. (Service: AmazonConfig; Status Code: 400; Error Code: MaxNumberOfConfigurationRecordersExceededException; Request ID: 4d48abd6-380a-4037-ab8e-51f239d203cc; Proxy: null).
```

Reason: Each AWS Region supports only one configuration recorder. CloudFormation StackSets will fail to create a stack instance in the account and Region if the recorder already exists. This happens when you're using AWS Config in that Region, or you used it in the past. For additional information, refer to [Configuration Recorder](#) in the *AWS Config Developer Guide*.

Resolution

Activate AWS Config in the appropriate Region and ensure that the necessary resource types are included in the recording group. For additional information, refer to [Enable AWS Config](#) in the *AWS WAF, AWS Firewall Manager, and AWS Shield Advanced Developer Guide*.

Problem: AWS Config isn't activated in member accounts

When AWS Config isn't activated in member accounts, you see following error message in your Firewall Manager console:

Accounts within policy scope (3)

Q Search by AWS account ID

AWS account ID	Status	Details
.....	⊗ Noncompliant	Missing required services: AWS Config. Details
.....	⊗ Noncompliant	Missing required services: AWS Config. Details
.....	⊗ Noncompliant	Missing required services: AWS Config. Details

Resolution

If you're using this solution's prerequisite template to activate AWS Config, then this is a transient issue. It takes time for AWS Config to activate and propagate across AWS Organizations accounts. Allow some time for the update to complete its processing. If you are not using this solution's prerequisite template, then access the individual accounts to activate AWS Config manually. For more information, refer to [Enable AWS Config](#) in the *AWS WAF, AWS Firewall Manager, and AWS Shield Advanced Developer Guide*.

Other errors

This section addresses other known errors when deploying or using this solution.

Problem: The FMS admin account-id isn't displayed in the Firewall Manager console

The Firewall Manager settings don't reflect the Admin account ID provided in the CloudFormation stack.

Resolution

It might take up to five minutes for the changes to update in the console.

Problem: The CloudFormation StackSets instance displays as Outdated

The CloudFormation StackSets instance displays an **Outdated** status.

eu-west-1	⊗ OUTDATED	User initiated operation
eu-west-2	⊗ OUTDATED	User initiated operation
eu-west-3	⊗ OUTDATED	User initiated operation
sa-east-1	⊗ OUTDATED	User initiated operation
us-east-2	⊗ OUTDATED	User initiated operation
us-west-1	⊗ OUTDATED	User initiated operation
us-west-2	⊗ OUTDATED	User initiated operation

Resolution

The **Outdated** status is temporary. Allow more time for the CloudFormation StackSets to update to a final state after the StackSets operation completes. Creating StackSets instances across multiple accounts and Regions is a time-intensive process. For example, for 6 accounts in approximately 18 Regions, it takes about 90 minutes to complete the StackSets operation.

Problem: InternalErrorException when creating a policy in Firewall Manager

Firewall Manager fails to create policies due to `InternalErrorException`.

```

▼ 2020-09-10T13:47:18.041-04:0... [ERROR] [fmsHelper/putPolicy] {"message":null,"code":"InternalErrorException","time":"2020-09-10
[ERROR] [fmsHelper/putPolicy]
{
  "message": null,
  "code": "InternalErrorException",
  "time": "2020-09-10T17:47:18.041Z",
  "requestId": "b8c75083-ec51-4cc0-a92c-5135abb1faf1",
  "statusCode": 400,
  "retryable": false,
  "retryDelay": 80.23010098902039
}

▼ 2020-09-10T13:47:18.041-04:0... [ERROR] [PolicyManager/saveShieldPolicy-Regional] failed to save policy
[ERROR] [PolicyManager/saveShieldPolicy-Regional] failed to save policy

```

Resolution

This issue is transient in nature, and invoking the Lambda function again fixes the issue. For example, after updating the `/FMS/Regions` parameter, follow the steps to invoke the update again. Use the following steps to invoke the event again:

1. Sign in to the [AWS Systems Manager console](#).
2. On the navigation menu, under **Application Management**, select **Parameter Store**.
3. Select the **/FMS/Regions** parameter and choose **Edit**.
4. Keep the default value and choose **Save changes**.

This invokes the `policyManager` Lambda function again using the same value. The Firewall Manager policy should successfully create.

Problem: Throttling exception with AWS APIs

AWS APIs throttling can occur if the solution is handling large number of Firewall Manager policies and AWS accounts. The following error is logged in CloudWatch logs:

```
[ERROR] [ComplianceGenerator/getComplianceDetails] ThrottlingException: Rate exceeded
```

Resolution

The Lambda functions include a `MAX_ATTEMPTS` environment variable, which you can [adjust](#) to fix this issue. The `MAX_ATTEMPTS` variable controls how many times the solution attempts to retry an API request.

Contact AWS Support

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

Create case

1. Sign in to [Support Center](#).
2. Choose **Create case**.

How can we help?

1. Choose **Technical**.

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

Additional information

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

Help us resolve your case faster

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

Solve now or contact us

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

Uninstall the solution

You can uninstall this solution from the AWS Management Console or by using the [AWS Command Line Interface](#) (AWS CLI).

Before uninstalling the solution, complete the following steps to ensure that the Firewall Manager security policies are deleted before the stack deletion:

1. Sign in to the [AWS Systems Manager console](#).
2. On the navigation menu, under **Application Management**, select **Parameter Store**.
3. Select the `/FMS/<Policy-Id>/OU` parameter and choose **Edit**.
4. Change the value to delete and choose **Save changes**.

All other resources deployed by this solution are automatically deleted when you delete the stack. Only custom defined rules are not automatically deleted.

Using the AWS Management Console

1. Sign in to the [CloudFormation console](#).
2. On the **Stacks** page, select this solution's installation stack.
3. Choose **Delete**.

Using AWS Command Line Interface

Determine whether the AWS CLI is available in your environment. For installation instructions, see [What Is the AWS Command Line Interface](#) in the *AWS CLI User Guide*. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

Note

This solution supports a complete deletion of the stack and all resources deployed by the solution. Only custom defined rules and an S3 bucket with compliance reports are left behind.

Deleting the Amazon S3 bucket

This solution is configured to retain the solution-created S3 bucket (for storing compliance reports) if you decide to delete the CloudFormation stack to prevent accidental data loss. After uninstalling the solution, you can manually delete this S3 bucket if you do not need to retain the data. Follow these steps to delete the Amazon S3 bucket.

1. Sign in to the [Amazon S3 console](#).
2. Choose **Buckets** from the left navigation pane.
3. Locate the *<stack-name>* S3 buckets.
4. Select the S3 bucket and choose **Delete**.

To delete the S3 bucket using AWS CLI, run the following command:

```
$ aws s3 rb s3://<bucket-name> --force
```


Use the solution

This section provides a user guide for using the AWS solution.

Set up the Systems Manager parameters

This solution uses three Systems Manager parameters to initiate creating, updating, and deleting Firewall Manager policies. Review the following scenarios for guidance to set up the following Systems Manager tasks:

- Create policies across two OUs and five AWS Regions
- Delete tags from policies
- Delete Regional policies
- Delete all policies

Each of the parameters is a *StringList* type. Use commas to separate each string.

Create policies across OUs and Regions

Use the following steps to create policies across two OUs and five AWS Regions with the scope of policies restricted to a certain tag value.

Note

For this example, we use the following values to represent variables:

- OUs: `ou-xxxx-y1y1y1y1`, `ou-yyyy-x2x2x2x2`
- Regions: `us-east-1`, `us-east-2`, `us-west-1`, `us-west-2`, `eu-west-1`
- Tag: `{"ResourceTags": [{"Key": "Environment", "Value": "Prod"}], "ExcludeResourceTags": false}`

1. Sign in to the [AWS Systems Manager console](#).
2. On the navigation menu, under **Application Management**, select **Parameter Store**.
3. Update the `/FMS/OUs` parameter:

- a. Select the **/FMS/OUs** parameter and choose **Edit**.
 - b. Update the parameter with the OU values. For this example, we use: `ou-xxxx-y1y1y1y1,ou-yyyy-x2x2x2x2`.
 - c. This action creates the Global AWS WAF and AWS Shield Advanced policies.
4. Update the **/FMS/Regions** parameter:
- a. Select the **/FMS/Regions** parameter and choose **Edit**.
 - b. Update the **/FMS/Regions** parameter with the chosen Regions. For this example, we use: `us-east-1,us-east-2,us-west-1,us-west-2,eu-west-1`.
 - c. This action creates the Regional policies (one AWS WAF, one AWS Shield, and two Security Groups).
5. Update the **/FMS/Tags** parameter:
- a. Select the **/FMS/Tags** parameter and choose **Edit**.
 - b. Update the **/FMS/Tags** parameter with the tag value. For this example, we use:

```
{"ResourceTags": [{"Key": "Environment", "Value": "Prod"}], "ExcludeResourceTags": false}.
```
 - c. This action updates all policies with the provided tag value.

The solution creates Firewall Manager after you complete these steps. Two global policies and four Regional policies should be in each of the selected Regions. In this scenario, 22 total policies are created, using the following formula:

(4 Regional policies × 5 Regions) + 2 global policies

Delete tags from policies

To delete tags from the policies, complete the following steps:

1. Sign in to the [AWS Systems Manager console](#).
2. On the navigation menu, under **Application Management**, select **Parameter Store**.
3. Select the **/FMS/Tags** parameter and choose **Edit**.
4. Update the **/FMS/Tags** parameter using the following value: `delete`

This action updates all policies and removes the applied tags.

Delete Regional policies

To delete all Regional policies, complete the following steps:

1. Sign in to the [AWS Systems Manager console](#).
2. On the navigation menu, under **Application Management**, select **Parameter Store**.
3. Select the **/FMS/Regions** parameter and choose **Edit**.
4. Update the **/FMS/Regions** parameter using the following value: delete

This action deletes all Regional policies.

Delete policies

To delete all policies, complete the following steps:

1. Sign in to the [AWS Systems Manager console](#).
2. On the navigation menu, under **Application Management**, select **Parameter Store**.
3. Select the **/FMS/OUs** parameter and choose **Edit**.
4. Update the **/FMS/OUs** parameter using the following value: delete

Note

The policy metadata is stored in the DynamoDB table. Don't delete this table while you're using the solution.

Access compliance reports

The `aws-fms-compliance.template` CloudFormation template deploys infrastructure needed to generate compliance reports on the Firewall Manager policies. This generates the following reports:

- **Account Compliance Report** – This report lists all member accounts in scope of the policy and their compliance status. You can find this report the [S3 bucket](#) with naming schema `<timestamp>_account_compliance_<policy-id>`.

MEMBER_ACCOUNT	COMPLIANCE_STATUS
.....	COMPLIANT
.....	COMPLIANT
.....	(*AWSCONFIG*: "Cannot create config rule resource for member account (.....). Please ensure AWS Config Recorder is enabled and the Config resource limits are not exceeded.")

- **Resource Violation Report** – This report lists all AWS resources in member accounts in scope of that policy, that are in violation of compliance. You can find this report can be in the S3 bucket with naming schema `<timestamp>_resource_violator_<policy-id>`.

MEMBER_ACCOUNT	RESOURCE_ID	RESOURCE_TYPE	VIOLATION_REASON
.....	AWS::EC2::SecurityGroup	RESOURCE_VIOLATES_AUDIT_SECURITY_GROUP
.....	AWS::EC2::SecurityGroup	RESOURCE_VIOLATES_AUDIT_SECURITY_GROUP

The S3 bucket that includes the reports has public access blocked, is encrypted, and has version turned on. Additionally, we recommend the following:

- Turning on multi-factor authentication (MFA) on object deletion for this bucket
- Ensuring that users don't gain elevated privileges to view or delete these reports (following the least privilege design principles).

For more information, refer to [Configuring MFA delete](#) in the *Amazon S3 User Guide*.

Access CloudWatch Logs insights

This solution logs error, warning, informational, and debugging messages for the Lambda functions. To choose the type of messages to log, locate the applicable function in the AWS Lambda console and change the **LOG_LEVEL** environment variable to the applicable type of message. For further instructions on how to change the variable, see [Using Lambda environment variables](#) in the *AWS Lambda Developer Guide*.

The following table lists the types of log levels you can choose from.

Level	Description
ERROR	Logs include information on anything that causes an operation to fail.
WARNING	Logs include information on anything that con potentially cause inconsistencies in the function but might not necessarily cause the

Level	Description
	operation to fail. Logs also include ERROR messages.
INFO	Logs include high-level information about how the function is operating. Logs also include ERROR and WARNING messages.
DEBUG	Logs include information that might be helpful when debugging a problem with the function. Logs also include ERROR, WARNING, and INFO messages.

You can adjust the log levels to troubleshoot the issues identified in [Troubleshooting](#).

Add CloudWatch Logs insights

Use the following procedure to add CloudWatch Logs insights to this solution.

1. Navigate to the [Amazon CloudWatch console](#).
2. On the navigation menu, under **Logs**, choose **Insights**.
3. On the **Logs Insights** page, choose the **Logs** tab.
4. Select **/aws/lambda/FMS-Stack-policyManager-~~xxxx~~**. This log group contains the log events related to policy creation, updates, and deletions.
5. Copy one of the following sample queries and paste it into the query field:
 - To identify error events:

```
fields @message
| parse @message "[*] [*] *" as loggingType, microService, loggingMessage
| filter loggingType = "ERROR"
| display loggingType, microService, loggingMessage
```

- To identify policy create success events:

```
fields @message
| parse @message "[*] [*] *" as loggingType, microService, loggingMessage
| filter loggingMessage like "FMS policy saved successfully"
```

```
| display loggingType, microService, loggingMessage
```

- To identify policy create fail events:

```
fields @message
| parse @message "[*] [*] *" as loggingType, microService, loggingMessage
| filter loggingMessage like "failed to save policy"
| display loggingType, microService, loggingMessage
```

6. Select a time preference and choose **Run query**. Save these queries for future use.

Developer guide

This section provides the source code for the solution, a [list of policies and rule sets](#), and [additional customizations](#).

Source code

Visit our [GitHub repository](#) to download the source files for this solution and to share your customizations with others.

The [AWS CDK](#) generates the solution templates. See the [README.md](#) file for additional information.

List of policies and rule sets

This section describes the policies and rule sets used with this solution.

Centralized WAF managed rules automation

To support Firewall Manager, this solution installs [AWS Managed Rules for AWS WAF](#). You can scope your accounts based on either OUs or resource tags.

The solution installs the following AWS Managed Rules:

- **Core Rule Set (CRS)– web ACL capacity unit (WCU) 700** – This group contains rules that are generally applicable to web applications. This group provides protection against exploitation of a wide range of vulnerabilities, including those described in [Open Worldwide Application Security Project](#) (OWASP) publications.
- **Amazon IP reputation list–WCU 25** – This group contains rules that are based on Amazon threat intelligence. This list is useful if you want to block sources associated with bots or other threats.
- **Known Bad Inputs (KBI)–WCU 200** – This group contains rules that allow you to block request patterns that are known to be not valid and are associated with exploitation or discovery of vulnerabilities. These inputs help reduce the risk of a malicious actor discovering a vulnerable application.
- **SQL–WCU 200** – This group contains rules that allow you to block request patterns associated with exploitation of SQL databases, like SQL injection attacks. These rules help prevent remote injection of unauthorized queries.

By default, any findings based on these rules are auto-remediated by Firewall Manager. You can change this setting to remediate manually by updating the selection in the solution's manifest file.

Centralized security group audit checks

In Firewall Manager, this solution installs pre-configured audit checks for VPC security groups in your Amazon EC2 instances across your accounts from a central admin account. You can scope the accounts based on either OUs or resource tags. The solution provides for auditing and cleanup of unused and redundant security groups.

By default, findings based on these rules are not auto-remediated by Firewall Manager.

Centralized DDoS protection enablement

If you activated AWS Shield Advanced, then you can leverage its rules and policies to protect from centralized DDoS attacks.

By default, findings based on these rules are auto-remediated by Firewall Manager. You can choose to change this setting to remediate manually by updating the selection in the solution's manifest file.

Centralized DNS Firewall rules automation

To support centralized management of DNS Firewall rules, the solution installs pre-configured DNS Firewall rule group in each Region. The DNS Firewall rule group uses [AWS Managed Domain Lists](#).

For more details, refer to [Route 53 Resolver DNS Firewall](#) in the *Amazon Route 53 Developer Guide*.

Policy manifest file

This solution uses a JSON manifest file to create Firewall Manager policies. When you deploy this solution, the manifest file is copied to an S3 bucket (`<Stack-Name>-<xx>-policymanifestbucket-<xx>`) in your account.

The manifest file is a set of opinionated defaults for the policies. If these defaults aren't suitable for your use case, you can adjust the configurations in the manifest by using the following sample policy manifest.



Sample policy manifest file

Manifest schema

Review the following schema details and definitions before updating the manifest file for your use case.

```

{
  "default": {
    "<Policy-Type>": <Policy-Object>
  }
}

```

- **default** – Manifest root key. **Do not** change.
- **Policy-Type** – Firewall Manager policies supported by the solution. The following list provides the supported types.
 - "WAF_GLOBAL"
 - "WAF_REGIONAL"
 - "SHIELD_GLOBAL", "SHIELD_REGIONAL"
 - "SECURITY_GROUPS_USAGE_AUDIT"
 - "SECURITY_GROUPS_CONTENT_AUDIT"

- "DNS_FIREWALL"
- **Policy-Object**
 - **policyName** – The name of the Firewall Manager policy.
 - **policyDetails** – Details about the policy that are specific to the service type, in JSON format. For details on different policy types, refer to [Security service policy data](#).
 - **resourceType** – The type of resource protected by or in scope of the policy. This is in the format shown in [AWS resource and property types reference](#).
 - **resourceTypeList** – A list of **resourceType**.
 - **remediationEnabled** – Indicates if the policy should be automatically applied to new resources and if the policy findings should be automatically remediated.

For further details on customizing the solution, refer to the [README.md](#) file in the GitHub repository.

Customization guide

This section provides customization instructions and examples for this solution.

Change the default Firewall Manager security policy configuration

This solution deploys Firewall Manager security policies with default configurations. However, you can change policy settings or apply different policies to different OUs and Regions.

To change the default Firewall Manager security policy configuration, follow these steps after [deploying the solution](#).

1. Sign in to the [Amazon S3 console](#).
2. Choose the `<Stack-Name>-<xx>-policymanifestbucket-<xx>` S3 bucket.
3. Choose the `policy_manifest.json` file in the bucket.
4. Download the manifest file and make adjustments to the default settings in the policy manifest. For more information, refer to [Policy manifest file](#).
5. Upload the updated manifest file in the same location.
6. [Update the Parameter Store parameters](#). After updating the parameters (OU, Region, or tag parameter), the Firewall Manager policies should also update to reflect the changes made in step 4.

Apply different policies to different OUs and Regions

To apply different policies to different OUs and Regions, follow these steps.

1. Use [aws-fms-policy.template](#) to launch additional resources needed to support different policies for different OUs and Regions. You can launch this template multiple times for as many policy configurations as needed.
2. Provide following stack parameter values:

Parameter	Default	Description
Policy Identifier	<Optional input>	A unique identifier for the policies.
Policy Table	<Optional input>	DynamoDB table where policy metadata will be saved. This table is created as part of primary template deployment .
UUID	<Optional input>	Universally unique identifier (UUID) for stack deployment. The UUID is created as part of primary template deployment . <div data-bbox="1081 1339 1510 1751" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>You can leave this parameter blank if you don't want to send an anonymized metric to the solution's endpoint.</p> </div>
Metric Queue	<Optional input>	Amazon Simple Queue Service (Amazon SQS) queue

Parameter	Default	Description
		<p>to send anonymized metrics to the solution endpoint. The queue is created as part of primary template deployment.</p> <div data-bbox="1081 478 1507 888" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p>Note</p> <p>You can leave this parameter blank if you don't want to send an anonymized metric to the solution's endpoint.</p> </div>

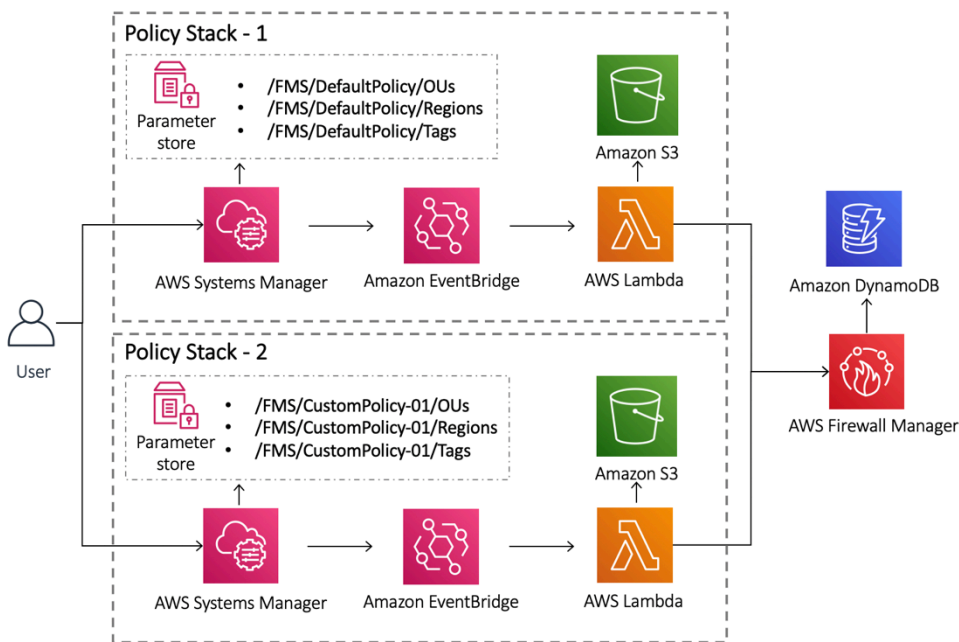
Note

Policy Table, UUID, and Metric Queue are created as part of the primary stack deployment. You can review their values by checking the [Outputs](#) section of the primary deployed stack. Ensure that you provide the same value as given in the **Outputs** section of the primary deployed stack.

- After the deployment succeeds, three more Parameter Stores are added in the Systems Manager console, as well as one more `<Stack-Name>-<xx>-policymanifestbucket-<xx>` bucket in the Amazon S3 console.

You can adjust these Parameter Store values. If you adjust them, the solution creates a Firewall Manager policy accordingly.

The policy configuration is managed by the `policy_manifest.json` file from the manifest bucket. You can update the `policy_manifest.json` file at any time. See [Policy manifest file](#) for more information.



Deploying multiple policy stacks for Firewall Manager

You can create as many policy stacks for different policy configurations as needed and apply them to different OUs and Regions.

Example policy customization scenarios

For details on policy manifest schema, refer to [Policy manifest file](#). You can configure the policy manifest in any number of ways. The following examples are some common scenarios.

Change policy auto-remediation behavior

All the policies have a default remediation behavior in the policy manifest file. You can adjust this as true or false per your requirements.

```
"remediationEnabled": false
```

Add AWS WAF Bot Control rule group

You can customize the **WAF Global** or **WAF Regional** policy in the manifest file to add AWS managed WAF Bot Control rule group. You can update the `preProcessRuleGroups` or `postProcessRuleGroups` section in the WAF policy as follows:

```
"postProcessRuleGroups": [{
```

```
"ruleGroupArn": null,
"overrideAction": {
  "type": "NONE"
},
"managedRuleGroupIdentifier": {
  "version": null,
  "vendorName": "AWS",
  "managedRuleGroupName": "AWSManagedRulesBotControlRuleSet"
},
"ruleGroupType": "ManagedRuleGroup",
"excludeRules": []
}]
```

For more information about the AWS WAF Bot Control managed rule group, refer to [AWS Managed Rules rule groups list](#) in the *AWS WAF, AWS Firewall Manager, and AWS Shield Advanced Developer Guide*.

Deploy specific policy types

You can deploy a selection of Firewall Manager policy from the supported policies:

- WAF_GLOBAL
- WAF_REGIONAL
- SHIELD_GLOBAL
- SHIELD_REGIONAL
- SECURITY_GROUPS_USAGE_AUDIT
- SECURITY_GROUPS_CONTENT_AUDIT
- DNS_FIREWALL

Each Firewall Manager policy type has a JSON object defined in the manifest schema that controls the policy configuration. You can remove this JSON object from the manifest file if you don't need a specific policy.

If the policy has already been created by the solution, use the following steps to delete a specific policy type:

1. Delete the deployed FMS policy type.
 - a. Sign in to the [AWS Firewall Manager console](#), using the admin account.

- b. Identify the policy to be deleted.
 - c. Select the policy and choose **Delete**.
 - d. Chose **Delete all policy resources** in the pop-up window, and choose **Delete**.
2. Update the policy manifest file in the S3 bucket. For more information, refer to [Policy manifest file](#).
3. Update Parameter Store parameters. For more information, refer to [Step 3. Add and manage Firewall Manager policies](#).

Reference

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

Anonymized data collection

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

- **Solution ID** - The AWS solution identifier
- **Unique ID (UUID)** - Randomly generated, unique identifier for each deployment
- **Timestamp** - Data-collection timestamp

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

1. Download the `aws-fms-prereq.template` [the section called "AWS CloudFormation template"](#) to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

```
"Mappings": {
  "PolicyStackMap": {
    "Metric": {
      "SendAnonymousMetric": "Yes"
    }
  },
```

to

```
"Mappings": {
  "PolicyStackMap": {
    "Metric": {
      "SendAnonymousMetric": "No"
    }
  },
```



```
},
```

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

Other AWS WAF solution and resources

- [AWS WAF Security Automations solution](#)

Contributors

- Garvit Singh
- Rakshana Balakrishnan
- Aijun Peng
- William Quan
- Nikhil Reddy
- Ryan Garay

Revisions

Date	Change
September 2020	Initial release
August 2021	Release version 2.0.0: Added support for DNS Firewall policies, generating compliance report on FMS policies, and multiple custom policy stack deployments. Also, migrated source code to <code>aws-sdk-js-v3</code> . For additional details, refer to the CHANGELOG.md file.
April 2022	Release version 2.0.1: Minor updates and bug fixes. For additional details, refer to the CHANGELOG.md file.
August 2022	Release version 2.0.2: Minor updates and bug fixes. For additional details, refer to the CHANGELOG.md file.
December 2022	Release version 2.0.3: Minor updates and npm vulnerability fixes. For additional details, refer to the CHANGELOG.md file.
April 2023	Release version 2.0.4: Fixed npm json5 vulnerabilities CVE-2022-46175 . Upgraded AWS CDK dependencies to version 2. And, mitigated impact caused by new default settings for S3 Object Ownership (ACLs disabled) for all new S3 buckets. For additional details, refer to the CHANGELOG.md file.
June 2023	Release version 2.0.5: Updated parameter names for consistency. For additional details, refer to the CHANGELOG.md file.

Date	Change
June 2023	Release version 2.0.6: Fixed dependabot issues for fast-xml-parser, CVE-2023-34104 . Fixed deployment issue which was limiting the solution to be deployed in only us-east-1. For additional details, refer to the CHANGELOG.md file.
August 2023	Release version 2.0.7: Updated aws-cdk-lib to 2.88 to force Lambda Node.js runtime update to Node 18.x and added AWS SDK updates to include newer version of fast-xml-parser. For additional details, refer to the CHANGELOG.md file.
October 2023	Release version 2.0.8: Updated package versions to resolve security vulnerabilities. For additional details, refer to the CHANGELOG.md file.
January 2024	Release version 2.0.9: Updated the Lambda function runtime to NodeJS 18.x. For additional details, refer to the CHANGELOG.md file.
February 2024	Documentation update: Applied new structure to implementation guide to improve flow and organization. The new structure includes a user guide and a developer guide that consolidate topics.
June 2024	Release version 2.0.10: Upgraded braces package to mitigate CVE-2024-4068 . Fixed intermittent deployment failure caused by CopyManifest custom resource installing latest SDK version. For additional details, refer to the CHANGELOG.md file.

Notices

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

Automations for AWS Firewall Manager is licensed under the terms of the [Apache License Version 2.0](#).