Automations for AWS
Firewall Manager
Implementation Guide
# Table of Contents

Solution overview ........................................................................................................ 1
Cost .................................................................................................................................. 2
  Cost per month for a small organization .................................................................. 2
  Cost per month for a large organization .................................................................. 3
Architecture overview .................................................................................................... 4
Solution components ...................................................................................................... 6
  AWS Lambda functions ............................................................................................. 6
  AWS CloudFormation StackSets ............................................................................... 6
  AWS Firewall Manager integration ......................................................................... 7
  AWS Systems Manager Parameter Store ............................................................... 7
  Amazon EventBridge ................................................................................................. 7
  Amazon Simple Storage Service .............................................................................. 7
  Amazon DynamoDB .................................................................................................. 7
Security .......................................................................................................................... 8
  IAM roles .................................................................................................................. 8
    Permissions required by the prerequisite stack .................................................... 8
    Permissions required by the primary stack .......................................................... 8
  AWS Systems Manager Parameter Store ............................................................... 8
Design considerations .................................................................................................. 10
  Deployments ............................................................................................................. 10
  Uninstalling the solution ......................................................................................... 10
AWS CloudFormation template .................................................................................... 11
Automated deployment .................................................................................................. 12
  Prerequisite ............................................................................................................. 12
  Deployment overview .............................................................................................. 12
  Step 1. Launch the stack ......................................................................................... 12
  Step 2. Add and manage FMS policies ................................................................... 13
    Access the Systems Manager Parameter Store history ....................................... 13
Additional resources ...................................................................................................... 15
Update the stack ............................................................................................................ 16
Scenarios for setting up the Systems Manager parameters ........................................... 17
  Scenario #1 .............................................................................................................. 17
  Scenario #2 .............................................................................................................. 17
  Scenario #3 .............................................................................................................. 18
  Scenario #4 .............................................................................................................. 18
List of policies and rule sets used ................................................................................... 19
  Centralized WAF managed rules automation ....................................................... 19
  Centralized security group audit checks ............................................................... 19
  Centralized DDoS protection enablement ............................................................. 19
  Centralized DNS Firewall rules automation ......................................................... 19
Policy manifest file ........................................................................................................ 20
  Manifest Schema ...................................................................................................... 20
Customize policies ....................................................................................................... 22
  Example policy customization scenarios ............................................................... 23
    Change policy auto-remediation behavior .......................................................... 24
    Add AWS WAF Bot Control rule group ............................................................. 24
    Deploy specific policy types .............................................................................. 24
Compliance reports ........................................................................................................ 26
Amazon CloudWatch logs insights .............................................................................. 27
  Add CloudWatch Logs insights ............................................................................. 27
Troubleshooting ........................................................................................................... 29
Common errors ............................................................................................................ 29
  Enabling AWS Config in Pre-Requisite stack does not work ................................. 29
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activating AWS Config using AWS CloudFormation StackSets fails when creating the configuration recorder</td>
<td>29</td>
</tr>
<tr>
<td>AWS Config is not activated in member accounts</td>
<td>30</td>
</tr>
<tr>
<td>The FMS admin account-id is not displayed in the Firewall Manager console</td>
<td>30</td>
</tr>
<tr>
<td>The AWS CloudFormation StackSets instance displays as Outdated</td>
<td>30</td>
</tr>
<tr>
<td>6. InternalErrorException when creating a policy in Firewall Manager</td>
<td>31</td>
</tr>
<tr>
<td>6. Throttling exception with AWS APIs</td>
<td>31</td>
</tr>
<tr>
<td>Install the prerequisite template</td>
<td>33</td>
</tr>
<tr>
<td>Step 1. Launch the prerequisite stack</td>
<td>33</td>
</tr>
<tr>
<td>Step 2. Manually activate AWS Firewall Manager</td>
<td>34</td>
</tr>
<tr>
<td>Uninstall the solution</td>
<td>36</td>
</tr>
<tr>
<td>Using the AWS Management Console</td>
<td>36</td>
</tr>
<tr>
<td>Using AWS Command Line Interface</td>
<td>36</td>
</tr>
<tr>
<td>Collection of operational metrics</td>
<td>37</td>
</tr>
<tr>
<td>Source code</td>
<td>38</td>
</tr>
<tr>
<td>Contributors</td>
<td>39</td>
</tr>
<tr>
<td>Revisions</td>
<td>40</td>
</tr>
<tr>
<td>Notices</td>
<td>41</td>
</tr>
<tr>
<td>AWS glossary</td>
<td>42</td>
</tr>
</tbody>
</table>
Centrally configure, manage, and audit firewall rules with Automations for AWS Firewall Manager

Publication date: September 2020 (last update (p. 40): August 2023)

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 Web Application Firewall (AWS WAF), and audit checks for VPC security groups and DNS Firewall rules 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 AWS Firewall Manager can be challenging and time consuming. To help simplify this process, this solution deploys a set of AWS managed firewall rules 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 on-boarding 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 their 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.

The supplemental AWS CloudFormation supplemental template included in this solution can be deployed in an AWS Organizations management account to configure the prerequisites for this solution automatically. For example:

- Checking that Organizations Full Feature is activated.
- Designating an account as the admin account for Firewall Manager.
- Optionally, automating the configuration of prerequisites, such as enabling AWS Config across an AWS Organization.

This implementation guide describes architectural considerations and configuration steps for deploying Automations for AWS Firewall Manager in the Amazon Web Services (AWS) Cloud. It includes links to an AWS CloudFormation template that launches and configures the AWS services required to deploy this solution using AWS best practices for security and availability. It is intended for IT administrators and DevOps professionals who have practical experience architecting in the AWS Cloud.
You are responsible for the cost of the AWS services used while running this solution. The total cost to run this solution depends on the number of policies installed, the number of rule sets and Web ACLs installed, the number and execution duration of AWS Lambda functions, and the number of Amazon EventBridge events published.

Prices are subject to change. For full details, refer to the pricing webpage for each AWS service you will be using in this solution.

The cost for each policy is **$100 per policy per month and per Region**. For example, for two Amazon CloudFront global policies and one Regional policy, the total policy cost is three policies x $100 = $300 per month. The significant factors influencing the total cost of running this solution are the number of accounts managed and the number of policies installed. As of the most recent revision, the cost to run the solution for a small organization in the US East (N. Virginia) Region is approximately **$1,733.00 per month**.

### Cost per month for a small organization

#### Assumptions:

- Accounts: 12 accounts across two organizational units (OUs)
- Number of AWS Regions: 3
- No subscription to AWS Shield Advanced
- Number of policies: 13
  - Amazon 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)
    - Route 53 Resolver DNS Firewall policy ($100 x 3 Regions)

#### AWS Service Components

<table>
<thead>
<tr>
<th>AWS Service</th>
<th>Components</th>
<th>Quantity</th>
<th>Accounts</th>
<th>$/month</th>
<th>Monthly Total [USD]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Firewall Manager</td>
<td>Policies</td>
<td>13</td>
<td>N/A</td>
<td>$100.00</td>
<td>$1,300.00</td>
</tr>
<tr>
<td></td>
<td>AWS WAF Web ACL</td>
<td>4</td>
<td>12</td>
<td>$5.00</td>
<td>$240.00</td>
</tr>
<tr>
<td></td>
<td>AWS WAF Rules</td>
<td>4*4</td>
<td>12</td>
<td>$1.00</td>
<td>$192.00</td>
</tr>
<tr>
<td>Other Services*</td>
<td>N/A</td>
<td>N/A</td>
<td>12</td>
<td>less than $1.00</td>
<td>$1.00</td>
</tr>
<tr>
<td>Total:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$1,733.00</td>
</tr>
</tbody>
</table>
Cost per month for a large organization

Assumptions:

- Accounts: 150 accounts across 20 OUs
- Number of AWS Regions: 10
- No subscription to AWS Shield Advanced
- 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)
  - Route 53 Resolver DNS Firewall policy ($100 x 10 Regions)

<table>
<thead>
<tr>
<th>AWS Service</th>
<th>Components</th>
<th>Quantity</th>
<th>Accounts</th>
<th>$/month</th>
<th>Monthly Total [USD]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS Firewall Manager</td>
<td>Policies</td>
<td>41</td>
<td>N/A</td>
<td>$100.00</td>
<td>$4,100.00</td>
</tr>
<tr>
<td>AWS WAF Web ACL</td>
<td>11</td>
<td>150</td>
<td></td>
<td>$5.00</td>
<td>$8,250.00</td>
</tr>
<tr>
<td>AWS WAF Rules</td>
<td>4 x 11</td>
<td>150</td>
<td></td>
<td>$1.00</td>
<td>$6,600.00</td>
</tr>
<tr>
<td>Other Services*</td>
<td>N/A</td>
<td>N/A</td>
<td>150</td>
<td>less than $1.00</td>
<td>$1.00</td>
</tr>
<tr>
<td>Total:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$18,951.00</td>
</tr>
</tbody>
</table>

*Note: Other AWS services include AWS Lambda, AWS EventBridge, AWS CloudFormation StackSets, AWS Config, Amazon Route 53 Resolver DNS Firewall, and AWS Systems Manager Parameter Store.

The cost estimates in the tables does not 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.

Prices are subject to change. For full details, refer to the pricing webpage for each AWS service you will be using in this solution.
Architecture overview

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

Automations for AWS Firewall Manager solution architecture

The architecture can be grouped into two separate workflows: Policy manager and Compliance report generator.

Policy manager

When the AWS 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.

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 (p. 17).

2. An Amazon EventBridge rule uses an event pattern to capture the System Manager parameter update event.

3. An Amazon 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 ACL consisting of AWS managed rule sets, DNS Firewall policies, and VPC security group audit policies. Additionally, if you have a subscription to...
**AWS Shield Advanced**, this solution deploys Shield Advanced policies to protect against Distributed Denial of Service (DDoS) attacks.

5. The PolicyManager Lambda function fetches the policy manifest file from the Amazon S3 bucket and uses the manifest file to create AWS Firewall Manager security policies.

6. AWS 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 (p. 17).

**Compliance report generator**

When the CloudFormation stack deploys, it creates a time-based Amazon CloudWatch Events rule, a Lambda function, a SNS topic, and an Amazon S3 bucket.

1. A time-based Amazon EventBridge rule invokes the Compliance Generator Lambda function.

2. The Compliance Generator Lambda fetches Firewall Manager policies in each Region and publishes the list of policy IDs in the SNS topic.

3. The SNS topic invokes the Compliance Generator Lambda function with the payload {PolicyId: string, Region: string}.

4. The compliance generator generates a compliance report for each of the policies and uploads the report in CSV format in an S3 bucket.
Solution components

AWS Lambda functions

This solution uses AWS Lambda functions to initiate prerequisite checks and the installation of policies and rule sets in organizational units (OUs) for AWS 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 management account
  - The AWS Organizations Full Feature option is activated
  - The AWS Firewall Manager admin is configured
  - Trusted access is activated between AWS Organizations and AWS CloudFormation StackSets
  - AWS Config is activated across AWS Organizations for all member accounts

Log information for this Lambda function is accessible from the Amazon CloudWatch console, under CloudWatch Logs, in the following Log Group: `/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 Amazon 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 also saves policy metadata in the DynamoDB table.

Log information for this Lambda function is accessible from the Amazon CloudWatch console, under CloudWatch Logs, in the following Log Group: `/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.

Log information for this Lambda function is accessible from the Amazon CloudWatch console, under CloudWatch Logs, in the following Log Group: `/aws/lambda/<Stack-Name>-xxx-ComplianceGenerator-xxx`

AWS CloudFormation StackSets

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

**Note**

The time taken to turn on AWS Config is dependent 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 six accounts and 16 Regions for two OUs.
AWS Firewall Manager integration

This solution automatically installs policies and rule sets for AWS Firewall Manager. By default, AWS WAF, security group and DNS Firewall security policies are installed. Additionally, if you have a subscription to AWS Shield Advanced, Shield policies are also installed.

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

AWS Systems Manager Parameter Store

AWS Systems Manager Parameter Store stores the solution's configuration parameters. These parameters can be used 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 AWS Lambda functions when updates are made to AWS Systems Manager 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 user).

Amazon Simple Storage Service

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

Amazon DynamoDB

This solution uses Amazon 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.
Security

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

IAM roles

AWS Identity and Access Management (IAM) roles allow customers to assign granular access policies and permissions to services and users in the AWS Cloud. This solution creates IAM roles that grant AWS 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 AWS Firewall Manager admin, and recording AWS Lambda events in Amazon CloudWatch Logs.

Permissions required by the primary stack

The appropriate IAM permissions are also required to manage Firewall Manager policies. These permissions include creating and deleting Firewall Manager policies for AWS WAF, AWS Shield, VPC Security Group and DNS Firewall, reading and writing Amazon DynamoDB tables with policy metadata, reading SSM Parameter information, and recording Lambda events in CloudWatch Logs. Additionally, the compliance generator Lambda function needs permission to describe all Firewall Manager policies, generate compliance reports, and upload them in an Amazon S3 bucket.

AWS Systems Manager Parameter Store

This solution uses AWS Systems Manager Parameter Store to initiate create, read, update, and delete (CRUD) operations to the Firewall Manager policies. SSM 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.

A 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 SSM 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 Amazon DynamoDB table and SSM Parameters resources.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ssm:GetParameter",
      "Effect": "Deny",
      "Sid": "DenyAccessToMyParameter"
    }
  ]
}
```
"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/OUs",
    "arn:aws:ssm:<region>::<account-id>:parameter/FMS/Regions",
    "arn:aws:ssm:<region>::<account-id>:parameter/FMS/Tags"
  ],
  "Effect": "Deny",
  "Sid": "FMSSMSecure"
]
Design considerations

Deployments

Although AWS Organizations and AWS Firewall Manager are available globally, both AWS services use us-east-1 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 any AWS Organizations service control policies or any 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-1 Region.

Uninstalling the solution

Before uninstalling the solution, we recommend that you navigate to the AWS Systems Manager Parameter Store and update the /FMS/OU parameter to delete. This ensures that the Firewall Manager security policies are deleted before the stack deletion. All other resources deployed by this solution are automatically deleted when you delete the stack. Only custom defined rules are not automatically deleted. For more information, refer to Uninstall the solution (p. 36).
AWS CloudFormation template

This solution uses AWS CloudFormation to automate the deployment of the Automations for AWS Firewall Manager solution in the AWS Cloud. It includes the following CloudFormation template, which you can download before deployment:

View template

aws-fms-automations.template: Use this template to launch the solution and all associated components. The default configuration deploys AWS Lambda functions, Amazon EventBridge, AWS Systems Manager Parameter Store, Amazon DynamoDB, Amazon S3 bucket and AWS Firewall Manager policies.
Automated deployment

Before you launch this solution, review the architecture, configuration, security, and other considerations discussed in this guide to figure out which installation method best suits your needs. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

**Time to deploy:** Approximately three minutes

Prerequisite

If you do not have AWS Firewall Manager configured in your AWS Organizations management account, then you must deploy the solution’s prerequisite template first. This template must be deployed in the Organizations management account with the Organizations Full Features option activated prior to deploying the template.

If Firewall Manager is already configured in your AWS Organizations management account, then you can skip the prerequisite template installation and proceed to the Deployment overview to install the `aws-fms-automations` template in your designated Firewall Manager administrator account.

**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.

For more information, refer to [Install the prerequisite template](p. 33).

Deployment overview

Use the following steps to deploy this solution on AWS. For detailed instructions, follow the links for each step.

**Step 1. Launch the stack**

1. Sign in to the AWS Management Console and use the button to the right to launch the `aws-fms-automations` AWS CloudFormation template.

   Alternatively, you can [download the template](#) as a starting point for your own implementation.

**Step 2. Add and manage FMS policies**
Step 2. Add and manage FMS policies

You can add AWS FMS policies across multiple OUs and Regions for your business needs. Using AWS 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 the left menu pane, 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 (p. 17).

Access the Systems Manager Parameter Store history

Take the following steps to identify the person that invoked a change to the parameters in the Systems Manager Parameter Store:
1. Sign in to the AWS Systems Manager console.
2. On the left menu pane, 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 Customize policies (p. 22). This section describes how you can use aws-fms-policy.template to apply a different set of policies to different OUs/Regions.
Additional resources

AWS services

- AWS Config
- AWS CloudFormation
- Amazon EventBridge
- AWS Firewall Manager
- AWS Lambda
- AWS Systems Manager
- AWS DynamoDB
- Amazon Simple Storage Service

Other AWS WAF solution and resources

- AWS WAF Security Automations solution
- AWS WAF Resources
Update the stack

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

1. Sign in to the AWS CloudFormation console, and deploy the aws-fms-automations template. Refer to Launch the stack (p. 12).
2. Once the stack is created successfully, modify the SSM Parameter Store parameters to create the FMS policies for your OUs, Regions, and tag values. Refer to Add and manage FMS policies (p. 13).
3. Additionally, you can configure the policies to meet custom requirements by changing values in the policy manifest file. For more information, refer to Customize policies (p. 22).
4. Once you have deployed your new FMS policies and ensured they are consistent with your requirements, delete the previously deployed version of the solution. Sign in to the AWS CloudFormation console, select the existing aws-centralized-waf-and-security-group-management CloudFormation stack and choose Delete. For more information, refer to Uninstall the solution (p. 36).

You have now safely migrated to the latest version of this solution and the supported FMS policies.
Scenarios for setting up the Systems Manager parameters

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

- Create policies across two organizational units (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.

Scenario #1

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

Scenario information:

- OUs: ou-xxxx-yl1y1y1, ou-yyyy-x2x2x2
- Regions: us-east-1, us-east-2, us-west-1, us-west-2, eu-west-1
- Tag: 
  
  ```json
  {"ResourceTags":
   [{"Key":"Environment","Value":"Prod"}],"ExcludeResourceTags":false}
  ```

1. Update the `/FMS/OUs` parameter with the OU values. ou-xxxx-yl1y1y1, ou-yyyy-x2x2x2
   
   This action creates the Global AWS WAF and AWS Shield Advanced policies.

2. Update the `/FMS/Regions` parameter with the chosen Regions. us-east-1, us-east-2, us-west-1, us-west-2, eu-west-1

   This action creates the Regional policies (one AWS WAF, one Shield, and two Security Groups).

3. Update the `/FMS/Tags` parameter with the tag value. 
   
   ```json
   {"ResourceTags":
    [{"Key":"Environment","Value":"Prod"}],"ExcludeResourceTags":false}
   ```

   This action updates all policies with the provided tag value.

AWS Firewall Manager policies are created after following 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: \((4*5) + 2\).

Scenario #2

To delete tags from the policies, update the `/FMS/Tags` parameter using the following value:
Scenario #3

To delete all Regional policies, update the /FMS/Regions parameter using the following value:

/FMS/Regions: delete

This action deletes all Regional policies.

Scenario #4

To delete ALL policies, update the /FMS/OUs parameter using the following value:

/FMS/OUs: delete

Note
The policy metadata is stored in the Amazon DynamoDB table. Do not delete this table while you are using the solution.
Centralized WAF managed rules automation

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

The following lists the AWS Managed Rules that are installed:

- **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 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 would like 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 AWS Firewall Manager. You can choose to change this setting to auto-remediate manually, by updating the selection in the solution’s manifest file.

Centralized security group audit checks

In AWS 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 AWS Firewall Manager.

Centralized DDoS protection enablement

If you have 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 AWS Firewall Manager. You can choose to change this setting to auto-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.
Policy manifest file

This solution uses a JSON manifest file to create Firewall Manager policies. When you deploy this solution, the manifest file gets copied to an Amazon 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 are not suitable for your use case, you can adjust the configurations in the manifest by using the following steps.

Sample policy manifest file

Manifest Schema

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

```json
{
  "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 AWS 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 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.
Customize policies

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

To change default Firewall Manager security policy configuration, follow the steps after installing the solution.

1. After the solution deployment succeeds, sign in to the Amazon S3 console and choose <Stack-Name>-XX-policymanifestbucket-xx S3 bucket.
2. Refer to the policy_manifest.json file in the bucket.
3. Download the manifest file and make adjustments to the default settings in the policy manifest. For more information, refer to Policy manifest file (p. 20). Upload the updated manifest file in the same location.
4. Update the SSM Parameter Store parameters. After updating the SSM parameters (OU, Region, or Tag parameter), the FMS policies should also get updated to reflect the changes made in Step 3.

To apply different policy to different OUs/Regions, follow the steps:

1. Use aws-fms-policy.template to launch additional resources needed to support different policies for different OUs/Regions. You can launch this template multiple times for as many policy configurations as needed.

2. Provide following stack parameter values:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy Identifier</td>
<td></td>
<td>A unique identifier for the policies.</td>
</tr>
<tr>
<td>Policy Table</td>
<td></td>
<td>DynamoDB table where policy metadata will be saved. This table is created as part of primary template deployment (p. 12).</td>
</tr>
<tr>
<td>UUID</td>
<td></td>
<td>Unique identifier for stack deployment. The UUID is created as part of primary template deployment (p. 12).</td>
</tr>
</tbody>
</table>

Note
This parameter can be left blank if you do not want to send an anonymized metric to the solution’s endpoint.
### Automations for AWS Firewall Manager Implementation Guide

#### Example policy customization scenarios

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric Queue</td>
<td>SQS queue to send anonymized metric to solution endpoint. The queue is created as part of primary template deployment (p. 12).</td>
<td></td>
</tr>
</tbody>
</table>

**Note**

*Policy Table, UUID, and Metric Queue* are created as part of the primary stack deployment and their values can be reviewed by checking the output section of the deployed stack. Ensure that you provide the same value as given in the output section of the primary stack.

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

4. You can adjust these SSM Parameter Store values and the FMS policy would get created as per your SSM Parameter Store values. Also, the policy configuration is managed by the `policy_manifest.json` file from the manifest bucket. The `policy_manifest` can be updated at any time.

---

**Deploying multiple policy stacks for AWS Firewall Manager**

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

---

**Example policy customization scenarios**

For details on policy manifest schema, refer to [Customize policies (p. 20)](#). The policy manifest can be configured in any number of ways and 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. This can be adjusted as true or false per requirement.

"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 rule group lists in the AWS WAF Developer Guide.

Deploy specific policy types

You can also deploy a selection of FMS policy from the supported policies:

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

Each FMS policy type has a JSON object defined in the manifest schema (p. 20) that controls the policy configuration. You can remove this JSON object from the manifest file if you do not 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. Log in to the AWS Firewall Manager 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 bucket. For more information, refer to Policy manifest file (p. 20).
3. Update SSM Parameter Store parameter. For more information, refer to Step 2. Add and manage FMS policies (p. 13).
Compliance reports

The aws-fms-compliance template deploys infrastructure needed to generate compliance reports on the Firewall Manager policies.

<table>
<thead>
<tr>
<th>MEMBER_ACCOUNT</th>
<th>COMPLIANCE_STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>COMPLIANT</td>
</tr>
</tbody>
</table>

1. **Account Compliance Report**: This report lists all member accounts in scope of the policy and their compliance status. The report can be found in the S3 bucket with naming schema `<timestamp>_account_compliance_<policy-id>`.

Sample Account Compliance Report

<table>
<thead>
<tr>
<th>MEMBER_ACCOUNT</th>
<th>RESOURCE_ID</th>
<th>RESOURCE_TYPE</th>
<th>VIOLATION_REASON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AWS::EC2::SecurityGroup</td>
<td>RESOURCE_VIOLATES_AUDIT_SECURITY_GROUP</td>
</tr>
</tbody>
</table>

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

Sample Resource Violator Report

The S3 bucket that includes the reports has public access blocked, is encrypted, and has version turned on. Additionally, we recommend turning on multi-factor authentication (MFA) on object deletion for this bucket and ensuring users do not 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.
Amazon CloudWatch logs insights

This solution logs error, warning, informational, and debugging messages for the AWS 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.

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>Logs will include information on anything that causes an operation to fail.</td>
</tr>
<tr>
<td>WARNING</td>
<td>Logs will include information on anything that can potentially cause</td>
</tr>
<tr>
<td></td>
<td>inconsistencies in the function but might not necessarily cause the</td>
</tr>
<tr>
<td></td>
<td>operation to fail. Logs will also include ERROR messages.</td>
</tr>
<tr>
<td>INFO</td>
<td>Logs will include high-level information about how the function is</td>
</tr>
<tr>
<td></td>
<td>operating. Logs will also include ERROR and WARN messages.</td>
</tr>
<tr>
<td>DEBUG</td>
<td>Logs will include information that might be helpful when debugging a</td>
</tr>
<tr>
<td></td>
<td>problem with the function. Logs will also include ERROR, WARNING, and</td>
</tr>
<tr>
<td></td>
<td>INFO messages.</td>
</tr>
</tbody>
</table>

You can adjust the log levels to troubleshoot the issues identified in Troubleshooting (p. 29).

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 left menu pane, under Logs, choose Insights.
3. On the Logs Insights page, choose the Logs tab.
4. Select `/aws/lambda/FMS-Stack-policyManager-xxxxx`. 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,
     ```
Add CloudWatch Logs insights

- To identify policy create fail events:

```
loggingMessage
| filter loggingMessage like "FMS policy saved successfully"
| display loggingType, microService, loggingMessage
```

```
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.
Troubleshooting

Prior to addressing the following common errors, you can adjust the level of detail in the Amazon CloudWatch Logs. For more details, refer to Amazon CloudWatch logs insights (p. 26).

Common errors

Enabling AWS Config in Pre-Requisite stack does not work

**Issue:** The following error occurs when you deploy solution's aws-fms-prereq.template, with 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-XBxz06V4RXl 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)

**Stack creation failed**

**Reason:** Trusted access for AWS CloudFormation StackSets can only be enabled using AWS CloudFormation Console. Refer to Enabling trusted access with AWS CloudFormation Stacksets in the AWS Organizations User Guide.

**Resolution:**

1. Go to the **AWS CloudFormation console** and choose Enable trusted access. (Providing a Registered Delegated Administrator is optional)

![Enable trusted access](enable_trusted_access.png)

2. Deploy the aws-fms-prereq.template again, and this time it should succeed.

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

**Issue:** The following error occurs in the StackSets console.

```
CloudFormation StackSets
```

Enable trusted access

2. Deploy the aws-fms-prereq.template again, and this time it should succeed.
AWS CloudFormation StackSets error message

Reason: Each AWS Region supports only one configuration recorder. AWS CloudFormation StackSets will fail to create a stack instance in the account/Region if the recorder already exists. This happens when you are 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 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.

AWS Config is not activated in member accounts

Issue: When AWS Config is not activated in member accounts, you will see following error message in your Firewall Manager console.

The FMS admin account-id is not displayed in the Firewall Manager console

Issue: FMS settings does not reflect the Admin account ID provided in the AWS CloudFormation stack.

Resolution: It may take up to five minutes for the changes to update in the console.

The AWS CloudFormation StackSets instance displays as Outdated

Issue: The AWS CloudFormation StackSets instance displays an Outdated status.
6. InternalErrorException when creating a policy in Firewall Manager

**Issue:** Firewall Manager fails to create policies due to InternalErrorException.

**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. Take the following steps to invoke the event again:

1. Navigate to the AWS Systems Manager Parameter Store console.
2. Select the `/FMS/Regions` parameter and choose **Edit**.
3. Keep the default value and choose **Save changes**.

The `policyManager` Lambda function is invoked again using the same value. The FMS policy should successfully create.

---

6. Throttling exception with AWS APIs

**Issue:** 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:
Resolution: The Lambda functions include a MAX_ATTEMPTS environment variable, which can be adjusted to fix this issue. The MAX_ATTEMPTS variable controls how many times the solution attempts to retry an API request.
Install the prerequisite template

Prerequisites (optional)

Installing the Firewall Manager prerequisite template in an AWS Organizations management 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 management account, an AWS Lambda function checks for the following prerequisites:

1. The AWS Organizations All Features 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.

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

To view the prerequisites for using AWS Firewall Manager, refer to Prerequisite (p. 12).

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

Step 1. Launch the prerequisite stack

This automated AWS CloudFormation template deploys the Firewall Manager prerequisite template in the AWS Cloud.
Automations for AWS Firewall Manager Implementation Guide

Step 2. Manually activate AWS Firewall Manager

Note
You are responsible for the cost of the AWS services used while running this solution. For more details, visit the Cost (p. 2) 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 use the button to the right to launch the aws-fms-prereq AWS CloudFormation template.

Alternatively, you can download the template as a starting point for your own implementation.

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

3. On the Specify stack details page, assign a name to your solution stack.

4. Under Parameters, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

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

5. Choose Next.

6. On the Configure stack options page, choose Next.

7. On the Review page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.

8. Choose Create stack to deploy the stack.

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

Step 2. Manually activate AWS Firewall Manager (optional)

Use the following procedure to manually activate AWS Firewall Manager in AWS Organizations.
1. Activate **AWS Organizations Full Feature**.
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*. 
Uninstall the solution

To uninstall this solution and ensure that all the policies are deleted, you must navigate to the AWS Systems Manager Parameter Store and change the /FMS/<Policy-Id>/OU parameter to delete. Then you can proceed with stack deletion using one of the following methods.

Using the AWS Management Console

1. Sign in to the AWS CloudFormation console.
2. Select this solution’s installation stack.
3. Choose Delete.

Using AWS Command Line Interface

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, refer to 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 <specify-your-stack-name>
```

**Note**

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

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 activated, the following information is collected and sent to AWS:

- **Solution ID**: The AWS solution identifier
- **Unique ID (UUID)**: Randomly generated, unique identifier for each solution 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 task.

Modify the AWS CloudFormation template mapping section from:

```json
"Send" : {
    "AnonymizedUsage" : { "Data" : "Yes" }
},
```

to:

```json
"Send" : {
    "AnonymizedUsage" : { "Data" : "No" }
},
```
Source code

Visit the solution’s GitHub repository to download the templates and scripts for this solution, and to share your customizations with others. Firewall Automation for Network Traffic on AWS templates are generated using the AWS Cloud Development Kit (AWS CDK). Refer to the README.md file for additional information.
Contributors

- Garvit Singh
- Rakshana Balakrishnan
- Aijun Peng
- William Quan
- Nikhil Reddy
# Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 2020</td>
<td>Initial release</td>
</tr>
<tr>
<td>August 2021</td>
<td>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 aws-sdk-js-v3. For additional details, refer to the CHANGELOG.md file.</td>
</tr>
<tr>
<td>April 2022</td>
<td>Release version 2.0.1: Minor updates and bug fixes. For additional details, refer to the CHANGELOG.md file.</td>
</tr>
<tr>
<td>August 2022</td>
<td>Release version 2.0.2: Minor updates and bug fixes. For additional details, refer to the CHANGELOG.md file.</td>
</tr>
<tr>
<td>December 2022</td>
<td>Release version 2.0.3: Minor updates and npm vulnerability fixes. For additional details, refer to the CHANGELOG.md file.</td>
</tr>
<tr>
<td>April 2023</td>
<td>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.</td>
</tr>
<tr>
<td>June 2023</td>
<td>Release version 2.0.5: Updated parameter names for consistency. For additional details, refer to the CHANGELOG.md file.</td>
</tr>
<tr>
<td>June 2023</td>
<td>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.</td>
</tr>
<tr>
<td>August 2023</td>
<td>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.</td>
</tr>
</tbody>
</table>
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 available at [The Apache Software Foundation](https://www.apache.org/licenses/LICENSE-2.0).
AWS glossary

For the latest AWS terminology, see the AWS glossary in the AWS Glossary Reference.