aws

Developer Guide

AWS Network Security Manager

AWS Network Security Manager: Developer Guide

Table of Contents

What is AWS Network Security Manager?

With AWS Network Security Manager, you can centrally define and deploy network security protections—such as AWS WAF and AWS Shield Advanced rules—across one or more accounts and resources in your organization. Instead of working with each resource individually, you define reusable policy objects (rules, templates, and policies) once and deploy them at scale, with continual monitoring to detect and remediate policy drift.

How AWS Network Security Manager works at a glance

You define a set of hierarchical policy objects and apply them to targeted scopes. The lowest level of policy objects are *rules*. You can group rules into *templates*, which you can combine into *policies* with priority-based conflict resolution. You then select accounts and resources to protect with a *scope*, and enforce your policies by publishing a *deployment*.

After deployment, AWS Network Security Manager continually monitors the scoped environment and deployed policies to check whether they match the intended definitions. If not, AWS Network Security Manager can optionally remediate resources that drift from the intended configuration. For full details on the resource model, lifecycle, and architecture, see *How AWS Network Security Manager works*.

Features of AWS Network Security Manager

- **Support for multiple firewall types** – AWS Network Security Manager supports AWS WAF for web ACL configurations and rule groups. Protected resource types include Amazon CloudFront distributions, Application Load Balancers, and Amazon API Gateway stages. AWS Network Security Manager also supports AWS Shield Advanced for DDoS protection of Amazon CloudFront distributions, Application Load Balancers, Classic Load Balancers, and Elastic IP addresses.
- **Reusable security configurations** – Define policy objects once as rules, group them into templates, and create scopes to target resources. Reuse rules, templates, and scopes across multiple policies and deployments.
- **Priority-based policy merging** – When multiple policies protect the same resource, AWS Network Security Manager merges their configurations based on policy priority, providing predictable conflict resolution.

- **Single-account and multi-administrator support** – Use AWS Network Security Manager in a single account or designate multiple administrator accounts, each with a priority level. When administrators' policies overlap, AWS Network Security Manager resolves conflicts by administrator priority first, then by policy priority within each account.
- **Draft and Active states** – Each AWS Network Security Manager resource, including rules, templates, policies, scopes, and deployments, supports Draft and Active states. Save resources as Draft without affecting live protections, then publish when ready to enforce them.
- **Versioned snapshots** – Each AWS Network Security Manager resource, including rules, templates, policies, scopes, and deployments, supports immutable, version-numbered snapshots. Use versioning to preserve known-good AWS Network Security Manager resource configurations.
- **Protected-resource synchronization diagnostics** – For each in-scope resource, see exactly which settings are out of sync and why, with expected settings compared to actual values for each configuration field.
- **Automatic remediation** – Optionally enable AWS Network Security Manager to automatically remediate out-of-sync resources and align them with the intended configuration without manual intervention.
- **Natural-language rule generation** – Use the `GenerateRuleConfiguration` operation to describe a rule in plain language and receive the corresponding configuration JSON, powered by Amazon Bedrock.

Use cases

The following are common use cases for AWS Network Security Manager:

- **Enforce baseline AWS WAF rules across all accounts** – Define a set of AWS WAF rule groups that every internet-facing application must use, then deploy them across your entire organization. AWS Network Security Manager automatically applies the rules to new in-scope resources as they come into scope.
- **Create specific custom or exception AWS WAF rules across specific resources** – Define a set of AWS WAF rules that apply only to a specific set of resources. This set can be a specific business unit or application that you identify by AWS account ID, OU, tag, or Amazon Resource Names (ARNs).
- **Verify DDoS protection on internet-facing resources** – Create an AWS Shield Advanced policy scoped to public resource types and deploy it organization-wide so that critical resources have DDoS protection enabled.

- **Detect and remediate configuration drift** – Use synchronization status monitoring to identify resources whose firewall configurations have drifted from the intended state, and optionally enable automatic remediation to bring them back into compliance.
- **Delegate security management to multiple teams** – Designate multiple administrator accounts with different priority levels and scoped access to specific accounts, organizational units, or firewall types, allowing teams to manage their own security policies within defined boundaries.

Related services

- **AWS WAF** – Use AWS WAF to protect your web applications and APIs against common web exploits and bots that might affect availability, compromise security, or consume excessive resources. AWS Network Security Manager uses AWS WAF to create and manage web ACLs and rule groups for resources in a single account or across accounts in your organization.
- **AWS Shield Advanced** – A managed DDoS protection service that provides expanded protections for your applications running on AWS. AWS Network Security Manager uses Shield Advanced to provide DDoS protections for resources in a single account or across accounts in your organization.
- **AWS Organizations** – Helps you centrally manage and govern your environment as you grow and scale your AWS resources. AWS Network Security Manager requires AWS Organizations to manage protections across member accounts.
- **AWS Config** – A service that enables you to assess, audit, and evaluate the configurations of your AWS resources. AWS Network Security Manager uses a service-linked AWS Config recorder to discover resources in your member accounts.

Accessing AWS Network Security Manager

You can access AWS Network Security Manager through the following interfaces:

AWS Management Console

A web-based interface for managing AWS Network Security Manager resources.

Open the AWS Network Security Manager console at https://console.aws.amazon.com/network-security-manager/.

AWS Command Line Interface (AWS CLI)

Provides commands for AWS Network Security Manager operations. For more information, see the AWS CLI Command Reference.

AWS SDKs

AWS provides SDKs that consist of libraries and sample code for various programming languages and platforms. The SDKs provide a convenient way to create programmatic access to AWS Network Security Manager. For more information, see Tools for Amazon Web Services.

AWS Network Security Manager API

A REST API that uses JSON over HTTPS. The API endpoint format is `network-security-manager.{region}.api.aws`. For more information, see the AWS Network Security Manager API Reference.

To get started with AWS Network Security Manager, see *Setting up AWS Network Security Manager* and *Tutorial: Deploy your first network security protection*.

Pricing

You pay for each in-scope resource that AWS Network Security Manager monitors and protects. Additional charges apply for the underlying security services that you deploy, such as AWS WAF and AWS Shield Advanced. For detailed pricing information, see AWS Network Security Manager pricing on the AWS website.

Are you a first-time user?

If you are a first-time user of AWS Network Security Manager, we recommend that you read the following topics in order:

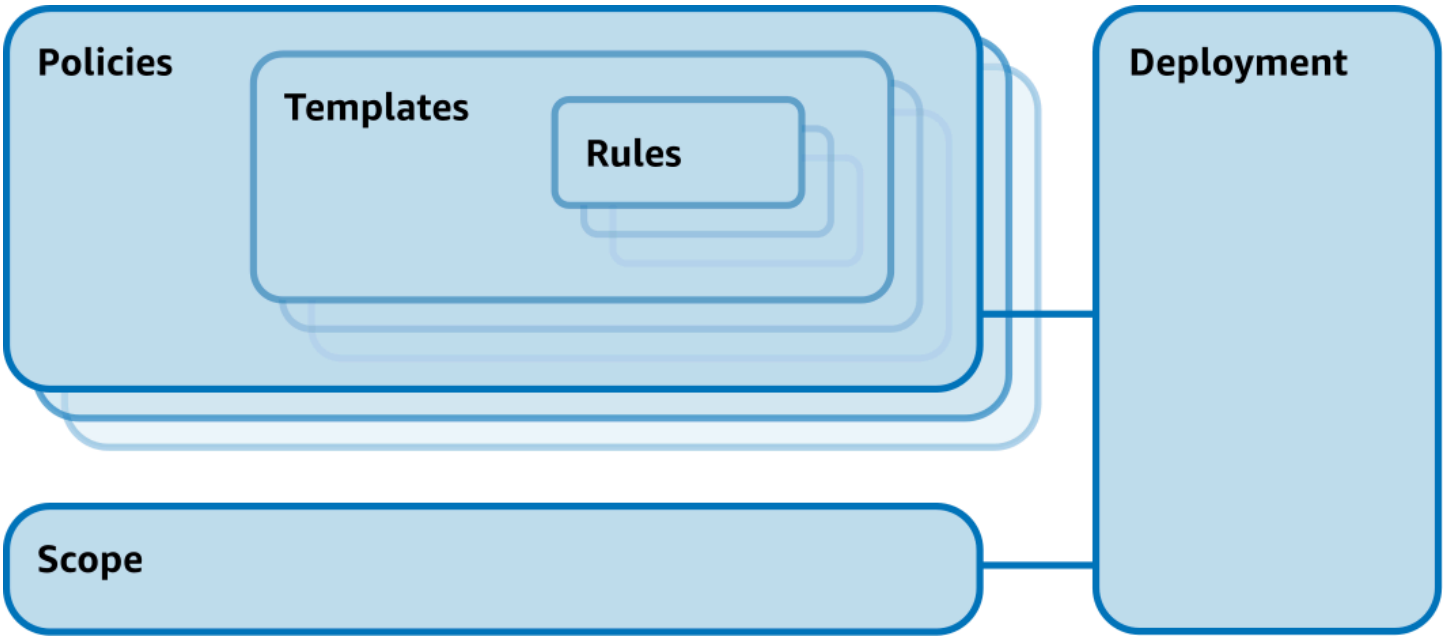
1. *How AWS Network Security Manager works* – Learn about the resource model, lifecycle states, policy merging, and synchronization.
2. *Setting up AWS Network Security Manager* – Complete the prerequisites for using AWS Network Security Manager.
3. *Tutorial: Deploy your first network security protection* – Walk through an end-to-end tutorial that creates your first deployment.

How AWS Network Security Manager works

Use the AWS Network Security Manager resource model to design and manage network security protections for resources in a single account or across multiple accounts in your organization. This chapter explains the five resource types, how they form a hierarchy, and the key concepts that govern their behavior.

AWS Network Security Manager resource model

AWS Network Security Manager uses five resource types arranged in a composition hierarchy:



Resources reference each other upward through the hierarchy. *Rules* are used within *templates* and *policies*. *Templates*, *rules*, or both are used within *policies*. A *scope* references the resources that *policies* protect within a *deployment*.

The following sections describe each resource type in the hierarchy.

Rule – The atomic unit of network security configuration. There are two types of rules – traffic inspection rules and firewall configuration rules. Traffic inspection focuses on how the security service addresses traffic. A configuration rule specifies system operations outside traffic inspection. The structure of a rule varies by firewall type.

Template – An ordered, reusable group of *rules* for a single firewall type. Use templates to package related rules together and share them across multiple *policies*. Templates are optional.

Policy – Combines an ordered set of *templates* and *rules* with enforcement settings for a firewall type. Each policy carries a priority (a lower number indicates a higher priority). When multiple policies protect the same resource, priority determines how conflicts resolve. For traffic inspection rules, priority determines the order in which rules are aggregated. For firewall configuration rules, priority determines which configuration applies for each resource and its in-scope policies.

Scope – A declarative selection of where protections apply. A scope specifies which resource types to protect, and you can optionally filter resources by tag or configuration attribute. In multi-account mode, a scope also specifies which accounts or OUs to include or exclude. In single-account mode, the scope applies to the account itself. A scope's mode is fixed when you create it. You can't switch a scope between multi-account and single-account modes in either direction, so a multi-account scope can never become a single-account scope, or the reverse.

Deployment – Binds one or more *policies* to a *scope*. Publishing a deployment activates enforcement. AWS Network Security Manager discovers resources in the scope, computes the intended configuration, and optionally remediates out-of-sync resources.

For resource composition limits, see *Quotas*.

AWS Network Security Manager resource lifecycle, merging, and synchronization

The following sections explain the behaviors and mechanisms that govern how AWS Network Security Manager resources operate after you create them.

Draft and Active lifecycle

Every AWS Network Security Manager resource – including rules, templates, policies, scopes, and deployments – has a status of either Draft or Active:

- **Active** – The default status. The resource is published and available for use. Other resources can reference it, and deployments that include it enforce its configuration.
- **Draft** – The resource is saved but not yet active. Draft resources are not enforced and cannot be referenced by other resources. When you create or update a resource, choose **Save draft** to keep it in draft status.

By default, AWS Network Security Manager creates resources as Active. To create a resource as Draft using the API, set the `isPublished` parameter to `false` in the create or update operation.

When you update a resource, the following scenarios apply:

Current state	Action	Result
Draft	Choose **Save draft**	Update the draft (remains Draft)
Draft	Choose **Publish**	Publish the draft (becomes Active)
Active	Choose **Publish**	Update and remain active (Active)
Active	Choose **Save draft**	Create a draft overlay (becomes Draft with an Active resource underneath)

The `hasPublishedVersion` field appears on list and get responses. This field tells you whether a published Active resource exists beneath a draft. When a resource has the Draft status and `hasPublishedVersion: true`, a pending draft sits on top of a live, published Active resource.

You cannot delete an Active resource that another Active resource references.

Versioning

Every AWS Network Security Manager resource – including rules, templates, policies, scopes, and deployments – can be versioned. To create a version of an Active resource, select the resource, choose **Action**, then choose **Create versioned snapshot**. Versions are not created automatically on every update.

> **AWS CLI alternative**
>
> You can also create a version using the `Create{Resource}Snapshot` API operation on an Active resource.

Key version properties:

- Each version has a unique number (a sequential integer) and a version-qualified ARN (for example, `arn:aws:network-security-manager:us-east-1:123456789012:rule:abc123:3`).
- Versions are immutable. They cannot be updated.
- You can create up to 10 versions per resource.
- Versions are taggable.

Use versions to preserve desired configuration variants. To roll back a resource, read a version, then update the live resource with the version's desired configuration.

Priority-based policy merging

When multiple policies protect the same resource, AWS Network Security Manager merges their configurations into a single result. A lower priority number indicates a higher priority (priority 1 takes precedence over priority 2). When AWS Network Security Manager merges policies that contain traffic inspection rules, priority determines the order in which the rules are aggregated. For firewall configuration rules, the higher-priority policy determines which configuration applies.

AWS Network Security Manager evaluates priority at two levels, in this order:

1. *Administrator priority* – AWS Network Security Manager evaluates administrator priority first. Each organization administrator is assigned a priority, as described in the section called "Administrator and member accounts". When AWS Network Security Manager merges policies, it ranks an administrator policy higher than a member account's own single-account policy. If an administrator policy and a member account's single-account policy both apply to a resource and conflict, the administrator policy's configuration wins. A member account cannot opt out of, override, or exempt itself from an administrator policy.
2. *Policy priority* – Within each account, AWS Network Security Manager then uses policy priority to resolve remaining conflicts. Policy priority values are integers starting from 1 and must be unique per account.

You set the administrator priority with the required `priority` field on the `PutAdminAccount` operation, an integer from 1 to 10. Priority values are unique, so two administrators never share a priority and no tie-breaking is required. For more information, see the AWS Network Security Manager API Reference.

The merged result of all applicable policies for a resource is the *effective firewall configuration*. AWS Network Security Manager computes this configuration, compares it to the resource's actual configuration, and reports any differences as synchronization status.

When policies conflict, AWS Network Security Manager compares each attribute and applies the value from the higher-priority policy. If the higher-priority policy does not define an attribute, the lower-priority policy value applies. A resource that matches the effective configuration is IN_SYNC. A resource protected by a lower-priority policy whose settings were overridden by a higher-priority policy reports as OUT_OF_SYNC for the lower-priority deployment. This is expected behavior, not a failure.

Resource discovery with AWS Config

AWS Network Security Manager discovers protected resources in your accounts using a service-linked AWS Config configuration recorder. This recorder is created automatically in each account when that account enters scope of a deployment. Unlike AWS Firewall Manager, you do not need to perform any additional setup. AWS Config usage by AWS Network Security Manager is not charged to the AWS Network Security Manager customer.

Key characteristics of the service-linked recorder:

- It is named `AWSConfigurationRecorderForNetworkSecurityManager`.
- It is separate from any customer-managed configuration recorder in the account.
- It records only the resource types relevant to your active scopes and deployments.
- You do not need to set up AWS Config or create a configuration recorder. AWS Network Security Manager manages this automatically. AWS Config usage by AWS Network Security Manager is not charged to the customer.
- You cannot modify or delete the recorder while it is in use. AWS Network Security Manager manages its lifecycle automatically.

You do not need to enable or configure AWS Config manually for AWS Network Security Manager to function. The service handles resource discovery automatically after you publish a deployment.

Synchronization and remediation

Enforcement in AWS Network Security Manager is asynchronous and eventually consistent. After you publish a deployment, or when a resource changes, the following stages occur:

1. **Resource discovery** – AWS Network Security Manager discovers resources in scope using a service-linked AWS Config recorder.
2. **Configuration computation** – For each discovered resource, AWS Network Security Manager computes the effective firewall configuration by merging all applicable policies by priority.
3. **Synchronization evaluation** – AWS Network Security Manager compares the effective configuration to the resource's actual configuration and records a synchronization status.
4. **Remediation** (optional) – If you set `remediationEnabled` on the policy, AWS Network Security Manager corrects out-of-sync resources to match the effective configuration.

Each stage is asynchronous. Consequences for your workflow:

- There is a delay between publishing a deployment and resources becoming protected.
- There is a delay between creating a new resource in scope and it being protected.
- The `Last Checked` timestamp on synchronization status entries tells you when AWS Network Security Manager last evaluated a resource.

Remediation and resource cleanup are policy-level settings. You configure each one per policy in the `policyConfiguration` object.

`remediationEnabled`

Specifies whether AWS Network Security Manager automatically remediates noncompliant resources. When you enable this setting, AWS Network Security Manager corrects out-of-sync resources to match the effective configuration.

`resourcesCleanUp`

Specifies whether AWS Network Security Manager automatically removes the resources it created when they are no longer needed. AWS Network Security Manager deletes a firewall it created when you remove the deployment, or when the account or resource goes out of scope.

The `remediationEnabled` and `resourcesCleanUp` settings both default to `false`. AWS Network Security Manager does not remediate or clean up resources until you enable these settings. The `policyConfiguration` object also holds `wafConfig`, which applies to AWS WAF policies. For more information, see the AWS Network Security Manager API Reference.

When a resource is out of sync, the synchronization status includes out-of-sync reasons with per-setting details that show expected values compared to actual values. The reasons are organized into the following categories:

- `missingFirewall` – No firewall protection is associated with the resource.
- `invalidFirewall` – A firewall exists but its configuration differs from the effective configuration. This includes six detail buckets: `incorrectSingleValueConfigurations`, `missingAppendableConfigurationValues`, `unexpectedAppendableConfigurationValues`, `incorrectAppendableConfigurationOrder`, `missingMergeableConfigurationValues`, and `unexpectedMergeableConfigurationValues`.

If remediation is enabled but AWS Network Security Manager cannot fix a resource, the `remediationIssues` field provides an `issueType`, a `message` describing the problem, and a `correctiveAction` suggesting how to resolve it.

AWS Network Security Manager does not evaluate settings that it does not manage. If a field does not appear in any contributing rule, AWS Network Security Manager ignores it during synchronization evaluation.

Administrator and member accounts

AWS Network Security Manager uses a cross-account management model built on AWS Organizations:

- **Administrator account** – The account where you create and manage AWS Network Security Manager resources (rules, templates, policies, scopes, and deployments). You designate one or more administrator accounts, each with a priority from 1 to 10.
- **Member accounts** – The accounts where AWS Network Security Manager discovers resources, evaluates synchronization status, and optionally remediates out-of-sync configurations. AWS Network Security Manager acts in member accounts through a service-linked role. For more information, see the section called “Service-linked role”.

You can optionally scope each administrator to a subset of accounts, OUs, and firewall types. Use this capability to delegate management of specific areas to different teams.

AWS Network Security Manager is a regional service. You create resources and deployments per Region with no cross-Region dependencies.

Service-linked role

AWS Network Security Manager uses an IAM service-linked role named `AWSServiceRoleForNetworkSecurityManager` to act in the member accounts of your organization on your behalf. This role trusts the `network-security-manager.amazonaws.com` service principal. It uses the `NetworkSecurityManagerServiceRolePolicy` AWS managed policy.

AWS Network Security Manager creates this role automatically when you enable the service, so there is no manual setup. The role lets AWS Network Security Manager manage AWS WAF and AWS Shield Advanced protections, discover protected resources through AWS Config, and perform cross-account operations. For more information, see the section called “Service-linked roles”.

Setting up AWS Network Security Manager

With AWS Network Security Manager, you can work in two organizational models. Use it directly in a single account, or use it with AWS Organizations to manage protections across multiple accounts. The setup steps depend on which model you choose.

Topics

Choose your organizational model

Choose from the following organizational models:

- **Single-account** – Use AWS Network Security Manager to protect resources in one account. You do not need AWS Organizations or an administrator account. Create an IAM identity with AWS Network Security Manager permissions and start using the service directly. AWS Network Security Manager automatically creates a service-linked role in your account.
- **Multi-account** – Use AWS Network Security Manager with AWS Organizations to protect resources across multiple accounts. This mode requires setting up AWS Organizations, enabling trusted access, and creating an administrator account.

Setting up for single-account use

To use AWS Network Security Manager in a single account, create an IAM identity with the appropriate permissions and start using the service directly. You do not need to create an administrator account. AWS Network Security Manager automatically creates a service-linked role in your account in the background. Skip to the section called “Create an IAM identity with AWS Network Security Manager permissions”.

Setting up for multi-account use with AWS Organizations

To manage protections across multiple accounts, establish the required AWS Organizations membership and configuration.

1. **Choose an account to use as the AWS Network Security Manager administrator for the organization** – This is the account from which you create and manage AWS Network Security Manager resources.
2. **Verify organization membership** – If your chosen account is not already a member of the organization, have it join. Follow the guidance at Inviting an AWS account to join your organization in the *AWS Organizations User Guide*.
3. **Enable all features in your organization** – AWS Organizations has two available feature sets: consolidated billing features and all features. To use AWS Network Security Manager, your organization must be enabled for all features. If your organization is configured only for consolidated billing, follow the guidance at Enabling all features in your organization in the *AWS Organizations User Guide*.
4. **Enable trusted access for AWS Network Security Manager** – Enable trusted access for AWS Network Security Manager in AWS Organizations. With trusted access enabled, AWS Network Security Manager can create a service-linked role in member accounts and manage resources on their behalf.
5. **Register a delegated administrator account** – Register your chosen account as a delegated administrator for AWS Network Security Manager in AWS Organizations. Use the Organizations management account to complete this step.

> ⚠ **Multi-account protections disabled when turning off trusted access**
>
> If you turn off trusted access for AWS Network Security Manager in AWS Organizations, or you deregister a delegated administrator account, AWS Network Security Manager sets the multi-account scope and deployment to a `DISABLED` state and cleans up the protections it manages.

Creating an AWS Network Security Manager administrator account

This step applies to multi-account mode only. Only the organization's management account can create AWS Network Security Manager administrator accounts.

After you complete the Organizations setup, register your account as an AWS Network Security Manager administrator. You can do this from the console or the AWS Command Line Interface (AWS CLI).

When you create an administrator account, AWS Network Security Manager automatically creates a service-linked role in all accounts in your organization. With this role, AWS Network Security Manager discovers resources and manages protections in member accounts.

With AWS Network Security Manager, you can have multiple administrators who manage the firewall resources of your organization. If you want to use multiple administrators in your organization, you can apply administrative scope conditions to each administrator to define the resources that they can manage. This gives you the flexibility to have different administrator roles within your organization and helps you maintain the principle of least privileged access. For example, one administrator can manage a set of organizational units (OUs). Another administrator can manage only specific policy types.

For the maximum number of administrators that you can have per organization, see *Quotas*.

Region registration

You register an AWS Network Security Manager administrator account in a single AWS Region. To use AWS Network Security Manager in additional Regions, repeat this step in each Region.

To create an administrator account using the console

1. Open the AWS Network Security Manager console at https://console.aws.amazon.com/network-security-manager/.
2. In the navigation pane, choose **Settings**.
3. Choose **Add administrator**.

4. Select the account and enter a priority value (1 to 10).
5. Choose **Add administrator** to confirm.

To create an administrator account using the AWS CLI

Call the `PutAdminAccount` operation. You must provide a priority value (1 to 10).

The following AWS CLI example registers an account as an AWS Network Security Manager administrator with priority 1:

```
aws network-security-manager put-admin-account \
    --account-id 123456789012 \
    --priority 1
```

You can designate multiple administrator accounts, each with a different priority. Optionally, you can limit each administrator to specific accounts, OUs, or firewall types by providing an `adminScope` parameter.

To remove an administrator account using the console

> ⚠ **Multi-account protections disabled when removing an administrator**
>
> When you remove an administrator account, AWS Network Security Manager sets the multi-account scope and deployment to a `DISABLED` state and cleans up the protections it manages.

1. Open the AWS Network Security Manager console at https://console.aws.amazon.com/network-security-manager/.
2. In the navigation pane, choose **Settings**.
3. Select the administrator account that you want to remove.
4. Choose **Remove administrator**.
5. To confirm, choose **Confirm administrator deletion**.

Create an IAM identity with AWS Network Security Manager permissions

To use AWS Network Security Manager, you need an IAM identity (user or role) with permissions to call AWS Network Security Manager operations. The IAM actions use the prefix `network-security-manager`.

The following example IAM policy contains three statements. The first statement grants full access to all AWS Network Security Manager operations. The second statement grants the additional AWS Organizations and AWS WAF permissions that AWS Network Security Manager needs. AWS Network Security Manager uses these permissions to identify the accounts and organizational units (OUs) in your organization and to list the available WAF rule groups and managed products. These permissions also let AWS Network Security Manager enable trusted access and register a delegated administrator in AWS Organizations. The third statement grants the `iam:CreateServiceLinkedRole` permission that AWS Network Security Manager needs to create its service-linked role, which it requires for both single-account and multi-account use.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "network-security-manager:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeOrganization",
                "organizations:ListAccounts",
                "organizations:ListRoots",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:ListAccountsForParent",
                "organizations:EnableAWSServiceAccess",
                "organizations:RegisterDelegatedAdministrator",
                "wafv2:ListRuleGroups",
                "wafv2:ListAvailableManagedRuleGroups",
                "wafv2:DescribeAllManagedProducts"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*"
        }
    ]
}
```

For production use, we recommend that you grant only the minimum permissions required. For more information, see the section called “Identity and access management”.

Sign up for an AWS account

To get started with AWS, you need an AWS account. For information about creating an AWS account, see Getting started with an AWS account in the *AWS Account Management Reference Guide*.

Using AWS Network Security Manager in Regions that are disabled by default

To use AWS Network Security Manager in a Region that is disabled by default, you must enable the Region for both the management account of your organization and the AWS Network Security Manager administrator account. For information about Regions that are disabled by default and how to enable them, see Managing AWS Regions in the *AWS General Reference*.

To enable a disabled Region

For both the Organizations management account and the AWS Network Security Manager administrator account, follow the guidance at Enabling a Region in the *AWS General Reference*.

Deploy your first network security protection

This tutorial walks you through deploying an AWS WAF web ACL using the AWS Network Security Manager console. You complete the following steps:

1. Create a rule
2. Create a template
3. Create a policy
4. Create a scope
5. Create and publish a deployment
6. Monitor synchronization status

In this tutorial, you create an AWS WAF web ACL that protects Application Load Balancers across your selected accounts. This process is largely identical for all security services and firewalls supported by AWS Network Security Manager.

Time to complete: approximately 10 minutes

Cost: This tutorial uses AWS services that might incur charges. For current pricing, see AWS Network Security Manager pricing on the AWS website.

Prerequisites

Before you begin, verify that you meet the following requirements:

- Completed setup steps as described in *Setting up AWS Network Security Manager*.
- An AWS Identity and Access Management (IAM) identity with AWS Network Security Manager permissions.
- An AWS Organizations organization with all features enabled and a designated AWS Network Security Manager administrator account (required for multi-account deployments).
- Access to the AWS Network Security Manager console.

Step 1: Create a rule

A rule is the building block of your security configuration. In this step, you create two AWS WAF rules: one that sets the default action and one that sets the visibility configuration. Default Action and VisibilityConfig are required fields for an AWS WAF web ACL.

To create the Default Action rule, follow these steps:

1. Open the AWS Network Security Manager console at https://console.aws.amazon.com/network-security-manager/.
2. In the navigation pane, choose **Overview**.
3. On the Overview page, choose **Create rule**.
4. For **Rule name**, enter a descriptive name, such as `my-default-action-rule`.
5. In the **Rule definition** section, for **Firewall type**, choose **AWS WAF**.
6. For **Select type of rule**, choose **Configuration**.
7. Choose **Insert config types**, then select **Default Action**. A JSON editor appears with a default configuration.
8. In the JSON editor, remove the `Block` section and keep only the `Allow` section.
9. Choose **Create rule**.

To create the VisibilityConfig rule, follow these steps:

1. In the navigation pane, choose **Overview**.
2. On the Overview page, choose **Create rule**.
3. For **Rule name**, enter a descriptive name, such as `my-visibility-config-rule`.
4. In the **Rule definition** section, for **Firewall type**, choose **AWS WAF**.
5. For **Select type of rule**, choose **Configuration**.
6. Choose **Insert config types**, then select **VisibilityConfig**. A JSON editor appears with a default configuration.
7. Choose **Create rule**.

AWS CLI alternative

You can also create rules with the AWS CLI:

```
aws network-security-manager create-rule \
    --rule-name "my-default-action-rule" \
    --firewall-type WAF \
    --rule-type CONFIGURATION \
    --configuration '{"DefaultAction": {"Allow": {}}}' \
    --is-published true

aws network-security-manager create-rule \
    --rule-name "my-visibility-config-rule" \
    --firewall-type WAF \
    --rule-type CONFIGURATION \
    --configuration '{"VisibilityConfig": {"SampledRequestsEnabled": true,
 "CloudWatchMetricsEnabled": true, "MetricName": "my-web-acl-metric"}}' \
    --is-published true
```

Step 2: Create a template

A template groups one or more rules for reuse across policies. In this step, you create a template that includes the rule from Step 1:

1. In the navigation pane, choose **Overview**.
2. On the Overview page, choose **Create template**.
3. For **Template name**, enter a descriptive name, such as `my-waf-template`.
4. For **Firewall type**, choose **AWS WAF**.
5. In the **Rules** section, choose **Add rule** and select the rules you created in Step 1.
6. Choose **Create template**.

AWS CLI alternative

You can also create a template with the AWS CLI:

```
aws network-security-manager create-template \
    --template-name "my-waf-template" \
    --firewall-type WAF \
    --associated-rule-list '[{"ruleIdentifier": "rule-arn-from-step-1"}]' \
```

```
    --is-published true
```

Step 3: Create a policy

A policy combines templates with enforcement settings. In this step, you create a policy that references your template and enables automatic remediation:

1. In the navigation pane, choose **Overview**.
2. On the Overview page, choose **Create policy**.
3. For **Policy name**, enter a descriptive name, such as `my-waf-policy`.
4. For **Firewall type**, choose **AWS WAF**.
5. In the **Policy priority** section, existing policies and their priorities display. Review the new policy's priority. If you need to adjust it, choose the priority field and change the value to 1 or any unused priority number.
6. In the **Templates and rules** section, choose **Add template** and select the template you created in Step 2.
7. For **Remediation**, select **Enabled**.
8. For **Existing customer web ACL resolution**, choose **No remediation**.
9. Choose **Create policy**.

AWS CLI alternative

You can also create a policy with the AWS CLI:

```
aws network-security-manager create-policy \
    --policy-name "my-waf-policy" \
    --firewall-type WAF \
    --priority 1 \
    --associated-template-and-rule-list '[{"templateIdentifier": "template-arn-
from-step-2"}]' \
    --policy-configuration '{"remediationEnabled": true, "resourcesCleanUp":
 false, "wafConfig": {"existingCustomerWebACLResolution": "NO_REMEDIATION"}}' \
    --is-published true
```

Step 4: Create a scope

A scope defines where your protections apply. In this step, you create a scope that targets Application Load Balancers in your accounts:

1. In the navigation pane, choose **Overview**.
2. On the Overview page, choose **Create scope**.
3. For **Scope name**, enter a descriptive name, such as `my-alb-scope`.
4. (Optional) In the **Account and organizational unit selection** section, for **Select method of controlling administrative scope**, select **Scope to entire AWS Organization (all OUs and accounts)**. Skip this step for single-account deployments.
5. In the **Resources** section, for **Resource types**, select **Application Load Balancer**.
6. Choose **Create scope**.

AWS CLI alternative

You can also create a scope with the AWS CLI:

```
aws network-security-manager create-scope \
    --scope-name "my-alb-scope" \
    --scope-configuration '{
        "accountFilter": {"includeAll": {}},
        "resourceScopes": {
            "AWS::ElasticLoadBalancingV2::LoadBalancer::application": {
                "includeAll": true
            }
        }
    }' \
    --is-published true
```

Step 5: Create and publish a deployment

A deployment binds policies to a scope and activates enforcement. To create and publish a deployment:

1. In the navigation pane, choose **Overview**.

2. On the Overview page, choose **Create deployment**.
3. For **Deployment name**, enter a descriptive name, such as `my-first-deployment`.
4. In the **Policies** section, choose **Add policy** and select the policy you created in Step 3.
5. In the **Scopes** section, choose **Add scope** and select the scope you created in Step 4.
6. Choose **Create deployment**.

AWS Network Security Manager begins discovering resources and applying protections. This process is asynchronous and might take a few minutes to complete.

AWS CLI alternative

You can also create and publish a deployment with the AWS CLI:

```
aws network-security-manager create-deployment \
    --deployment-name "my-first-deployment" \
    --associated-policy-list '[{"policyIdentifier": "policy-arn-from-step-3"}]'
 \
    --associated-scope-list '[{"scopeIdentifier": "scope-arn-from-step-4"}]' \
    --deployment-configuration '{"enableCrossAccountVisibility": true}' \
    --is-published true
```

Step 6: Monitor synchronization status

After you publish a deployment, AWS Network Security Manager discovers resources and synchronizes their configurations. To monitor the status of each resource:

1. In the navigation pane, choose **Overview**.
2. Choose the **Deployments** tab.
3. Choose the deployment you created in Step 5.
4. Choose the **Synchronization** tab.
5. Review the synchronization status for each resource:
   - **IN_SYNC** – The resource configuration matches your intended protection settings.
   - **OUT_OF_SYNC** – The resource configuration does not match. AWS Network Security Manager remediates this automatically when remediation is enabled.

- **NOT_APPLICABLE** – The resource is in scope but cannot be evaluated by this deployment.

AWS CLI alternative

You can also check synchronization status with the AWS CLI:

```
aws network-security-manager list-resource-synchronization-statuses \
    --deployment-identifier "deployment-arn-from-step-5"
```

Step 7: Clean up

If you no longer need the resources you created, delete them in reverse order to stop enforcement and remove protections from in-scope resources:

The reverse order matters because these resources reference each other. To see what is associated with a resource before you delete it, call `ListResourceAssociations` with that resource's ARN. It returns each associated resource's ARN and resource type. For more information, see the AWS Network Security Manager API Reference.

1. In the navigation pane, choose **Overview**. Choose the **Deployments** tab. Select your deployment, choose **Action**, then choose **Delete**.
2. Choose the **Scopes** tab. Select your scope, choose **Action**, then choose **Delete**.
3. Choose the **Policies** tab. Select your policy, choose **Action**, then choose **Delete**.
4. Choose the **Templates** tab. Select your template, choose **Action**, then choose **Delete**.
5. Choose the **Rules** tab. Select your rule, choose **Action**, then choose **Delete**.

AWS CLI alternative

Before you delete a resource, check its associations to plan the delete order:

```
aws network-security-manager list-resource-associations \
    --resource-arn "resource-arn"
```

If a resource still has associations, remove or delete the associated resources first. You can delete resources with the AWS CLI in reverse order:

```
aws network-security-manager delete-deployment \
    --deployment-identifier "deployment-arn"

aws network-security-manager delete-scope \
    --scope-identifier "scope-arn"

aws network-security-manager delete-policy \
    --policy-identifier "policy-arn"

aws network-security-manager delete-template \
    --template-identifier "template-arn"

aws network-security-manager delete-rule \
    --rule-identifier "rule-arn"
```

Next steps

Now that you have a working deployment, explore the following ways to expand your network security protection:

- Learn about the resource model and lifecycle in *How AWS Network Security Manager works*.
- Use the draft/publish workflow to stage changes before they take effect. For more information, see the section called “Draft and Active lifecycle”.
- Create snapshots to preserve known-good configurations. For more information, see the section called “Versioning”.
- Use `GenerateRuleConfiguration` to create rules from natural-language descriptions. For more information, see the AWS Network Security Manager API Reference.
- Explore managing AWS WAF and AWS Shield Advanced protections across your organization. For more information, see *Managing firewalls and security services*.

Managing firewalls and security services

With AWS Network Security Manager, you define and deploy security policy for multiple firewalls and security services. Although common elements span the AWS network security portfolio, important variations exist that you need to address within AWS Network Security Manager.

For each supported network security service, the following subsections describe the unique elements and steps for both a Quick Create and a standard detailed creation process.

Managing web ACLs for AWS WAF

With AWS Network Security Manager, you centrally configure and deploy AWS WAF web access control lists (web ACLs) across accounts and resources in your organization. You can define AWS Network Security Manager rules for AWS WAF, group them into protection templates, and enforce them at scale through AWS Network Security Manager policies and deployments.

AWS WAF concepts

The following concepts are specific to using AWS Network Security Manager with AWS WAF. For more information about how AWS Network Security Manager works, see *How AWS Network Security Manager works*.

AWS WAF rule types

For AWS WAF, a rule is one of two types:

- **INSPECTION** – Contains a rule group definition using `PreProcessFirewallManagerRuleGroups` (rules that run before customer-defined rules) or `PostProcessFirewallManagerRuleGroups` (rules that run after).
- **CONFIGURATION** – Contains a single web ACL configuration setting. The configuration JSON must contain exactly one top-level key matching a supported setting (for example, `Description`, `DefaultAction`, `VisibilityConfig`, `LoggingConfiguration`, `CaptchaConfig`, `ChallengeConfig`, `CustomResponseBodies`, `DataProtectionConfig`, `AssociationConfig`, `OnSourceDDoSProtectionConfig`, `TokenDomains`, or `MonetizationConfig`).

Most web ACL configuration settings correspond to parameters of the AWS WAF `CreateWebACL` operation. For more information about these settings and their definitions, see CreateWebACL in the *AWS WAF API Reference*.

The exception is `LoggingConfiguration`, which you configure separately. For more information, see the section called "Logging".

AWS WAF policy merging

For AWS WAF, each configuration setting uses one of the following merge strategies:

SingleValue

Settings that can have only one value. The highest-priority policy wins. Priority values are unique, so two policies cannot define different values at the same priority. For more information about how AWS Network Security Manager resolves priority, see the section called "Priority-based policy merging". The following settings use SingleValue: `DefaultAction`, `VisibilityConfig`, `CaptchaConfig`, `ChallengeConfig`, `OnSourceDDoSProtectionConfig`, `LoggingConfiguration`, `MonetizationConfig`, and `Description`.

Append (and AppendAndSort)

Entries from all policies append to the list in priority order. `TokenDomains` and `DataProtectionConfig.DataProtections` use Append. `PreProcessFirewallManagerRuleGroups` and `PostProcessFirewallManagerRuleGroups` use AppendAndSort, which appends rule groups from all policies and sorts them by priority. Order matters because AWS WAF evaluates rule groups sequentially.

Merge

Entries from all policies merge into a single set, regardless of priority. `CustomResponseBodies` and `AssociationConfig.RequestBody` use Merge.

The Append, AppendAndSort, and Merge strategies remove duplicate entries. When two policies contribute identical configuration, AWS Network Security Manager keeps the entry from the higher-priority policy. The effective configuration therefore contains one copy of that entry, not one copy per contributing policy.

Existing web ACL resolution

Use the `existingCustomerWebACLResolution` setting on an AWS WAF policy to specify how AWS Network Security Manager handles resources that already have a customer-managed web ACL:

- `RETROFIT` – Update your existing web ACL to align with the effective configuration.
- `OVERRIDE_ASSOCIATION` – Associate a web ACL managed by AWS Network Security Manager with the resource instead.
- `NO_REMEDIATION` – Do not remediate if your web ACL exists.

Logging web ACL traffic

You control web ACL logging with the `LoggingConfiguration` setting on a policy. AWS Network Security Manager applies this setting using the AWS WAF `PutLoggingConfiguration` operation. You can define one logging destination per web ACL. For more information, see PutLoggingConfiguration in the *AWS WAF API Reference*.

Every in-scope web ACL that the policy manages sends its logs to the destination that the policy specifies. This gives you centralized logging across the accounts in your organization. Use it to get detailed information about the traffic that your web ACLs analyze organization-wide.

For centralized logging across multiple accounts, you can send logs to an Amazon S3 bucket or an Amazon Data Firehose delivery stream. Each destination type requires additional configuration, which you must complete before AWS Network Security Manager can manage AWS WAF logging across your in-scope resources and accounts. The destination name must begin with `aws-waf-logs-`.

Amazon Data Firehose

Create the delivery stream with the following characteristics:

- Created using your AWS Network Security Manager administrator account
- A name that starts with `aws-waf-logs-`, for example `aws-waf-logs-network-security-manager-central`
- A PUT source
- Created in the Region where you operate, or in the US East (N. Virginia) Region if you capture logs for Amazon CloudFront

Before you use the delivery stream, test it to confirm that it has enough throughput for your organization's logs. For more information, see Creating an Amazon Data Firehose delivery stream in the *Amazon Data Firehose Developer Guide*.

Amazon S3

Create the bucket following the Amazon S3 guidance in the *AWS WAF Developer Guide*. You must also configure the bucket with the permissions that allow log delivery to write to it. The following bucket policy grants those permissions.

```
{
    "Version":"2012-10-17",
    "Id": "AWSLogDeliveryForNetworkSecurityManager",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheckNetworkSecurityManager",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::aws-waf-logs-bucket-suffix"
        },
        {
            "Sid": "AWSLogDeliveryWriteNetworkSecurityManager",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::aws-waf-logs-bucket-suffix/policy-id/AWSLogs/
*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

Replace *bucket-suffix* with your own bucket name suffix, and replace *policy-id* with the ID of your policy.

CloudWatch Logs supports single-account use only

If you use AWS Network Security Manager across multiple accounts, you cannot use an Amazon CloudWatch Logs log group as the logging destination. The log group must be in the same AWS account and the same Region as the web ACL. For multi-account deployments, use an Amazon S3 bucket or an Amazon Data Firehose delivery stream instead.

Quick Create for AWS WAF

With Quick Create, you define and deploy AWS WAF protections in a single guided workflow. Use Quick Create when you want to set up a complete web ACL protection from rules through deployment in one session.

To define and deploy an AWS WAF protection with Quick Create, complete the following procedures in order.

To create the policy and add protections

1. Open the AWS Network Security Manager console at https://console.aws.amazon.com/network-security-manager/.
2. In the navigation pane, choose **Overview**.
3. In the top right of the **Overview** page, choose **Quick create and deploy**. The **Quick create and deploy** page appears with multiple panes.
4. In the **Policy details** pane, for **Name**, enter a policy name. (Optional) For **Description**, enter a description.
5. In the **Define protections** pane, choose the firewall type that you want to work with. Select **AWS WAF**. A **Templates and rules** sub-pane appears, where you can add rules or templates.
6. To add rules, choose **Add or manage rules**. In the window that appears, for **Select rule type**, choose one of the following:
   - **AWS Managed Rules** – The table shows all of the AWS Managed Rules that AWS WAF supports.
   - **Rules** – The table shows the AWS WAF rule groups that you own, other than the AWS Managed Rules.

Select the check boxes for the rules that you want to add to your policy. Then choose **Add rules**.

7. To add templates, choose **Add templates**. In the **Add templates** window that appears, the table displays any available templates. Select the templates that you want, along with the version to include in the policy. Then choose **Add templates**.

To configure the policy

1. Review your selections in the **Templates and rules** table. The integer in the **Position** column determines the evaluation order of the rules and templates. To change the evaluation order, choose the position value in the **Position** column. Use the arrows to change the value. To accept the change, choose **Confirm**. To revert the change, choose **Cancel**.
2. Under **Policy configuration** in the **Define protections** pane, decide the automated actions that AWS Network Security Manager takes in the targeted scope:
   - **Remediation** applies the defined policy to any resources whose security state is not synchronized with the desired state that is described in AWS Network Security Manager.
   - **Resource cleanup** deletes the firewall that AWS Network Security Manager created for a resource when the deployment is removed, or when the account or resource is out of scope.
   - **Existing customer WebACL resolution** determines how to resolve conflicts between the policy that you are defining and what might already be deployed. From the menu, choose one of the following options:
     - **No remediation** skips remediation on any resources that already have a web ACL.
     - **Override association** replaces the existing web ACL with the AWS Network Security Manager created web ACL.
     - **Retrofit** retrofits its rules onto each existing web ACL and keeps the web ACL associated with the resource.
3. In the **Policy priority** pane, set the priority. AWS Network Security Manager evaluates policies according to their priority to resolve conflicts. Lower positions (lower integers) have a higher priority. To edit the priority, choose the **Position** value. Then increment it with the arrows, or enter a value directly. To confirm the selection, choose **Confirm**. To discard the change, choose **Cancel**.

To set the scope

1. In the **Set the scope** pane, define the set of assets that the policies apply to. For **Scope name**, enter a name. (Optional) For **Description**, enter a description.
2.

> **Multi-account mode only**
>
> This step applies to multi-account mode only. In multi-account mode, use the **Account and organizational unit selection** sub-pane to choose which accounts or OUs to include or exclude. In single-account mode, this sub-pane does not appear in the console, so skip this step.

   In the account and organizational unit selection sub-pane, select one of the following options:

   - Scope to the entire AWS Organization. No further definition is required.
   - Scope to the entire AWS Organization but exclude specific OUs and accounts. An **Organizational units** sub-pane and an **Accounts** sub-pane appear. To exclude organizational units, choose **Add OUs**. Select the organizational units in the organization tree, and then choose **Add OUs**. To exclude individual accounts, choose **Add accounts**. Select the accounts in the organization tree, and then choose **Add accounts**.
   - Do not scope to the entire AWS Organization, and only include specific OUs and accounts. Use the same **Add OUs** and **Add accounts** mechanism to include the organizational units and accounts that you want.
3. In the **Resources** sub-pane, for **Resource types**, select all or specific AWS WAF resource types. The options are one or more of the following:

   - **Application Load Balancer**
   - **Amazon API Gateway REST API**
   - **CloudFront Distribution**

> **CloudFront Region requirement**
>
> To protect **CloudFront Distribution** resources, use the US East (N. Virginia) Region (`us-east-1`).

> **Global and regional resource types**
>
> You can select the global resource type (**CloudFront Distribution**) or the regional resource types (**Application Load Balancer** and **Amazon API Gateway REST API**), but not both. After you select a resource type, you can't switch between **CloudFront Distribution** and the regional resource types when you update the configuration.

> **Unmanaged CloudFront resources**
>
> AWS Network Security Manager doesn't manage CloudFront distributions that use the CloudFront flat-rate pricing plan or CloudFront staging distributions. These resources appear out of synchronization.

4. For each selected resource type, choose one of the following mutually exclusive options:
   - **Include all resources**
   - **Include specific resources**
   - **Exclude specific resources**
5. If you choose **Include specific resources**, choose one of the following expression types, and then build the appropriate combinations of tags or ARNs:
   - **Single criteria**
   - **Match all criteria (AND)**
   - **Match any criteria (OR)**
   - **NOT**

   All resource types have the **Tag** and **ARN** inclusion criteria. If you choose **Tag**, enter a **Key** and an optional **Value**. You can identify multiple tags. If your resource type is an **Application Load Balancer**, an additional **Configuration** criterion appears. Within **Configuration**, choose a **Scheme**, which has the options **Internet-facing** or **Internal**.

To review and deploy the policy

1. In the **Review and deploy** pane, for **Deployment name**, enter a name. (Optional) For **Description**, enter a description. To determine whether the content of your deployment is visible to other accounts in your organization, select **Enable cross-account visibility for this deployment**.
2. (Optional) To add tags to your resource, choose **Add new tag**, and then enter **Key** and optional **Value** combinations.
3. To run what you defined, choose **Create and deploy**. When the process is complete, the **Deployments** page appears, where you can see the details of what was created.

> **Deployments might incur charges**
>
> A deployment creates network security services and attaches them to a protected resource. AWS Network Security Manager charges for protected resources. The security services themselves protect the resource, which incurs additional charges. Before you deploy, review the AWS Network Security Manager pricing page and the AWS WAF pricing page.

Standard creation process for AWS WAF

With the standard creation process, you create each AWS WAF component—rules, templates, policies, scopes, and deployments—individually for full control. Use this approach when you want to create reusable components and combine them as needed.

An AWS Network Security Manager AWS WAF policy can contain rules and templates. When you apply the policy, AWS Network Security Manager creates a web ACL in each in-scope account.

To create a rule (console)

1. Sign in to the AWS Management Console using your AWS Network Security Manager administrator account, and then open the AWS Network Security Manager console at https://console.aws.amazon.com/network-security-manager/.

> **Administrator account setup**
>
> For information about setting up an AWS Network Security Manager administrator account, see *Setting up AWS Network Security Manager*.

2. In the navigation pane, choose **AWS Network Security Manager**, then **Overview**.
3. In the **Overview** pane, choose **Create rule**. Enter a descriptive name, and then enter an optional **Description**.
4. In the **Rule definition** pane, for **Firewall type**, choose **AWS WAF**.
5. For **Type of rule**, choose either **Inspection** or **Configuration**.

   If you choose **Inspection**, a JSON editor pane named **Define rule group reference** appears with the following options:

   - **Insert template** – Select a JSON template for **First (pre-process)** rules or **Last (post-process)** rules. When you make a selection, AWS Network Security Manager inserts the corresponding JSON into the editor, along with warning icons if inputs are required to complete the template.
   - **Insert rule group** – Presents a selection list driven by three options:
     - **An AWS Managed Rules (AMR) rule group**
     - **An existing custom rule group**
     - **An AWS Marketplace rule group**

     AWS Network Security Manager presents the rule groups associated with your choice, and you can select a single instance for inclusion. After you make the selection, choose **Select rule group**. AWS Network Security Manager inserts references to the inspection rule with the correct syntax into the JSON editor.
   - **Insert with AI** – Opens an overlay window named **AI configuration generator**. Use this tool to describe your rule in plain language, and AWS Network Security Manager generates the associated JSON in the editor. You can then review the generated code and choose to accept, modify, or delete the output.
6. (Optional) After your JSON rule definition is complete, you can associate a tag with your rule. Choose **Add new tag**, and then enter a **Key** and an optional **Value**.
7. When you finish your definitions, choose either **Save draft** to save an inactive, potentially incomplete object, or **Create rule**.

> ⚠ **Web ACL names are permanent**
>
> You cannot change a web ACL name after you create it. If you update your policy's name, AWS Network Security Manager does not update the associated web ACL name. To have

AWS Network Security Manager create a web ACL with a different name, you must create a new policy.

To create a template (console)

A template assembles rules in priority order for evaluation within a policy. You can organize your rule types into groupings that align with your security goals. You can also let different teams build security control statements that align with their responsibilities and skills.

1. On the **Overview** page, in the **Overview** pane, choose **Create template**. On the **Create template** page, enter a **Template name**, enter an optional **Description**, and choose a **Firewall type** (in this case, **AWS WAF**).
2. When you choose a **Firewall type**, the **Rules** pane appears. Choose one of the two **Add rules** buttons to open a list of available rules. Select the check box to the left of a rule **Name** to add it to the template. To choose a specific version of a rule, select the **Version** field of the rule. If multiple versions are available, select the version you want, and then choose **Confirm**.
3. (Optional) Create a tag for the template. When you finish, choose either **Save draft** or **Create template**. To continue to create a policy, choose **Create template**.

To create a policy (console)

This is the final step of policy definition and preparation for deployment.

1. On the **Overview** page, in the **Overview** pane, choose **Create policy**. On the **Create policy** page, enter a **Name** and an optional **Description**, and then choose a **Firewall type** (in this case, **AWS WAF**).
2. In the **Policy priority** pane, set the priority. AWS Network Security Manager evaluates policies according to their priority to resolve conflicts. Lower positions (lower integers) have a higher priority. To edit the priority, choose the **Position** value. Then increment it with the arrows, or enter a value directly. To confirm the selection, choose **Confirm**. To discard the change, choose **Cancel**.
3. In the **Policy configuration** pane, decide the automated actions that AWS Network Security Manager takes in the targeted scope:
   - **Remediation** applies the defined policy to any resources whose security state is not synchronized with the desired state that is described in AWS Network Security Manager.

- **Resource cleanup** deletes the firewall that AWS Network Security Manager created for a resource when the deployment is removed, or when the account or resource is out of scope.
- **Existing customer WebACL resolution** determines how to resolve conflicts between the policy that you are defining and what might already be deployed. From the menu, choose one of the following options:
  - **No remediation** skips remediation on any resources that already have a web ACL.
  - **Override association** replaces the existing web ACL with the AWS Network Security Manager created web ACL.
  - **Retrofit** retrofits its rules onto each existing web ACL and keeps the web ACL associated with the resource.

4. After these configuration settings are complete, select the rules and templates that you want to add to the policy. In the **Templates and rules** pane, choose **Add rules** or **Add templates** to populate the policy. Both processes open a window that operates the same way as the rule selection process described earlier in this section.

To create a scope (console)

1. You can start this step from the **Overview** page or from other areas within the tool. In the **Scope details** pane, enter a **Scope name** and an optional **Description**.
2. **Multi-account mode only**

   This step applies to multi-account mode only. In multi-account mode, use the **Account and organizational unit selection** sub-pane to choose which accounts or OUs to include or exclude. In single-account mode, this sub-pane does not appear in the console, so skip this step.

   In the **Account and organizational unit selection** pane, choose one of three methods for controlling administrative scope:

   - **Scope to entire AWS Organization (all OUs and accounts)** – No further definition is required.
   - **Scope to entire AWS Organization but exclude specific OUs and accounts** – Two sub-panes appear, one for **Organizational units** and one for **Accounts**. Choose **Add OUs** to open the **Add OUs** window, select the organizational units you need from the organization tree, and then

choose **Add OUs**. To add individual accounts, in the **Accounts** sub-pane, choose **Add accounts**, select the accounts from the organization tree, and then choose **Add accounts**.

- **Do not scope to entire AWS Organization, only include specific OUs and accounts** – Use the same sub-panes to add OUs or accounts with the same mechanism described for the preceding option.

3. Next, select resources in the **Resources** sub-pane. In the **Resource types** selection box, select all or specific AWS WAF resource types. For AWS WAF, the options are any or all of the following:

   - **Application Load Balancer**
   - **Amazon API Gateway REST API**
   - **CloudFront Distribution**

   > **CloudFront Region requirement**
   >
   > To protect **CloudFront Distribution** resources, use the US East (N. Virginia) Region (`us-east-1`).

   > **Global and regional resource types**
   >
   > You can select the global resource type (**CloudFront Distribution**) or the regional resource types (**Application Load Balancer** and **Amazon API Gateway REST API**), but not both. After you select a resource type, you can't switch between **CloudFront Distribution** and the regional resource types when you update the configuration.

   > **Unmanaged CloudFront resources**
   >
   > AWS Network Security Manager doesn't manage CloudFront distributions that use the CloudFront flat-rate pricing plan or CloudFront staging distributions. These resources appear out of synchronization.

4. For each selected resource type, choose one of the following mutually exclusive options:

   - **Include all resources**
   - **Include specific resources**
   - **Exclude specific resources**

5. If you choose **Include specific resources**, select one of the following expression types:
   - **Single criteria**
   - **Match all criteria (AND)**
   - **Match any criteria (OR)**
   - **NOT**
6. Each expression type supports single or multiple inclusion criteria and ARNs, so you can build combinations of tags or ARNs. All resource types support the **Tag** and **ARN** inclusion criteria. If you choose **Tag**, identify a **Key** and an optional **Value**. You can identify multiple tags.
7. For an **Application Load Balancer**, an additional **Configuration** criterion appears. Within **Configuration**, you can set a **Scheme** with the options **Internet-facing** or **Internal**.

To create a deployment (console)

A deployment binds your policies to a scope and activates enforcement, which causes AWS Network Security Manager to apply your AWS WAF rules to the in-scope resources.

1. On the **Overview** page, in the **Overview** pane, choose **Create deployment**. You can also start this step from other areas within the tool.
2. Choose the policy that you want to deploy.
3. Choose the scope that defines the accounts and resources where AWS Network Security Manager applies the policy.
4. For **Deployment name**, enter a name. (Optional) For **Description**, enter a description.
5. (Optional) To make the deployment visible to other accounts in your organization, select **Enable cross-account visibility for this deployment**.
6. (Optional) To add tags, choose **Add new tag**, and then enter **Key** and optional **Value** combinations.
7. Choose **Create and deploy**. The **Deployments** page appears, where you can see the details of what AWS Network Security Manager created.

> ⚠ **Deployments might incur charges**
>
> A deployment creates network security services and attaches them to a protected resource. AWS Network Security Manager charges for protected resources. The security services

themselves protect the resource, which incurs additional charges. Before you deploy, review the AWS Network Security Manager pricing page and the AWS WAF pricing page.

Managing DDoS protections for AWS Shield Advanced

With AWS Network Security Manager, you centrally configure and deploy AWS Shield Advanced protections across accounts and resources in your organization. You can define DDoS protection policies and enforce them at scale through AWS Network Security Manager policies and deployments.

AWS Network Security Manager is billed separately

AWS Network Security Manager is not a part of the AWS Shield Advanced offer. For more information about pricing, see AWS Network Security Manager pricing and AWS Shield Advanced pricing.

AWS Shield Advanced subscription

When an active deployment includes an account in its scope, AWS Network Security Manager subscribes that account to AWS Shield Advanced. AWS Network Security Manager does not cancel the subscription when the account leaves the scope or when you remove the deployment. To cancel an AWS Shield Advanced subscription, see Subscribing to AWS Shield Advanced in the *AWS WAF Developer Guide*.

Quick Create for AWS Shield Advanced

With Quick Create, you define and deploy AWS Shield Advanced protections in a single guided workflow. Use Quick Create when you want to set up DDoS protection from policies through deployment in one session.

To define and deploy an AWS Shield Advanced protection with Quick Create, complete the following procedures in order.

To create the policy and configure protections

Complete the following steps:

1. Open the AWS Network Security Manager console at https://console.aws.amazon.com/network-security-manager/.
2. In the navigation pane, choose **Overview**.
3. On the **Overview** page, choose **Quick create and deploy**. The **Quick create and deploy** page appears with multiple panes.
4. In the **Policy details** pane, for **Name**, enter a policy name. (Optional) For **Description**, enter a description.
5. In the **Define protections** pane, choose the firewall type that you want to work with. Select **AWS Shield Advanced**.
6. Under **Policy configuration** in the **Define protections** pane, decide the automated actions that AWS Network Security Manager takes in the targeted scope:
   - **Remediation** applies the defined policy to any resources whose security state does not match the desired state that AWS Network Security Manager describes.
   - **Resource cleanup** deletes the DDoS protection that AWS Network Security Manager created for a resource when the deployment is removed, or when the account or resource is out of scope.
   - **Existing customer DDoS resolution** determines how to resolve conflicts between the policy that you are defining and what might already be deployed. From the menu, choose one of the following options:
     - **No remediation** skips remediation on any resources that already have a DDoS protection.
     - **Override association** replaces the existing DDoS protection with the AWS Network Security Manager created DDoS protection.
     - **Retrofit** retrofits its rules onto each existing DDoS protection and keeps the DDoS protection associated with the resource.
7. In the **Policy priority** pane, set the priority. AWS Network Security Manager evaluates policies according to their priority to resolve conflicts. Lower positions (lower integers) have a higher priority.

   To edit the priority, choose the **Position** value. Then increment it with the arrows, or enter a value directly. To confirm the selection, choose **Confirm**. To discard the change, choose **Cancel**.

To set the scope

Complete the following steps:

1. In the **Set the scope** pane, define the set of assets that the policies apply to. For **Scope name**, enter a name. (Optional) For **Description**, enter a description.
2. **Multi-account mode only**

   This step applies to multi-account mode only. In multi-account mode, use the **Account and organizational unit selection** sub-pane to choose which accounts or OUs to include or exclude. In single-account mode, this sub-pane does not appear in the console, so skip this step.

   In the account and organizational unit selection sub-pane, select one of the following options:

   - Scope to the entire AWS Organization. No further definition is required.
   - Scope to the entire AWS Organization but exclude specific OUs and accounts. An **Organizational units** sub-pane and an **Accounts** sub-pane appear. To exclude organizational units, choose **Add OUs**. Select the organizational units in the organization tree, and then choose **Add OUs**. To exclude individual accounts, choose **Add accounts**. Select the accounts in the organization tree, and then choose **Add accounts**.
   - Do not scope to the entire AWS Organization, and only include specific OUs and accounts. Use the same **Add OUs** and **Add accounts** mechanism to include the organizational units and accounts that you want.
3. In the **Resources** sub-pane, for **Resource types**, select all or specific AWS Shield Advanced resource types. The options are one or more of the following:
   - **Application Load Balancer**
   - **CloudFront Distribution**
   - **Classic Load Balancer**
   - **Elastic IP address**
4. For each selected resource type, choose one of the following mutually exclusive options:
   - **Include all resources**
   - **Include specific resources**
   - **Exclude specific resources**
5. If you choose **Include specific resources**, choose one of the following expression types, and then build the appropriate combinations of tags or ARNs:

- **Single criteria**
- **Match all criteria (AND)**
- **Match any criteria (OR)**
- **NOT**

All resource types have the **Tag** and **ARN** inclusion criteria. If you choose **Tag**, enter a **Key** and an optional **Value**. You can identify multiple tags.

> **CloudFront Region requirement**
>
> To protect **CloudFront Distribution** resources, use the US East (N. Virginia) Region (`us-east-1`).

To review and deploy the policy

Complete the following steps:

1. In the **Review and deploy** pane, for **Deployment name**, enter a name. (Optional) For **Description**, enter a description. To determine whether the content of your deployment is visible to other accounts in your organization, select **Enable cross-account visibility for this deployment**.
2. (Optional) To add tags to your resource, choose **Add new tag**, and then enter **Key** and optional **Value** combinations.
3. To create and deploy the policy and scope, choose **Create and deploy**. When the process is complete, the **Deployments** page appears, where you can see the details of what AWS Network Security Manager created.

> **Deployments might incur charges**
>
> A deployment creates network security services and attaches them to a protected resource. AWS Network Security Manager charges for protected resources. The security services themselves protect the resource, which incurs additional charges. Before you deploy, review the AWS Network Security Manager pricing page and the AWS Shield Advanced pricing page.

Standard creation process for AWS Shield Advanced

With the standard creation process, you set up each AWS Shield Advanced component—policies, scopes, and deployments—individually for full control. Use this approach when you want to create reusable components and combine them as needed.

An AWS Shield Advanced policy is simple and does not contain rules, rule groups, or templates. When you apply the policy, AWS Network Security Manager creates an AWS Shield Advanced DDoS protection in each in-scope account. For information about how AWS Network Security Manager works, see *How AWS Network Security Manager works*.

To create a policy (console)

Complete the following steps:

1. Sign in to the AWS Management Console using your AWS Network Security Manager administrator account, and then open the AWS Network Security Manager console at https://console.aws.amazon.com/network-security-manager/.

   > **Administrator account setup**
   >
   > For information about setting up an AWS Network Security Manager administrator account, see *Setting up AWS Network Security Manager*.

2. In the navigation pane, choose **AWS Network Security Manager**, then **Overview**.
3. In the **Overview** pane, choose **Create policy**. Enter a **Name** and an optional **Description**, and then choose a **Firewall type** (in this case, **AWS Shield Advanced**).
4. In the **Policy priority** pane, set the priority. AWS Network Security Manager evaluates policies according to their priority to resolve conflicts. Lower positions (lower integers) have a higher priority.

   To edit the priority, choose the **Position** value. Then increment it with the arrows, or enter a value directly. To confirm the selection, choose **Confirm**. To discard the change, choose **Cancel**.
5. In the **Policy configuration** pane, decide the automated actions that AWS Network Security Manager takes in the targeted scope:
   - **Remediation** applies the defined policy to any resources whose security state does not match the desired state that AWS Network Security Manager describes.

- **Resource cleanup** deletes the DDoS protection that AWS Network Security Manager created for a resource when the deployment is removed, or when the account or resource is out of scope.
- **Existing customer DDoS resolution** determines how to resolve conflicts between the policy that you are defining and what might already be deployed. From the menu, choose one of the following options:
  - **No remediation** skips remediation on any resources that already have a DDoS protection.
  - **Override association** replaces the existing DDoS protection with the AWS Network Security Manager created DDoS protection.
  - **Retrofit** retrofits its rules onto each existing DDoS protection and keeps the DDoS protection associated with the resource.

To create a scope (console)

Complete the following steps:

1. In the **Scope details** pane, enter a **Scope name** and an optional **Description**.
2. **Multi-account mode only**

   This step applies to multi-account mode only. In multi-account mode, use the **Account and organizational unit selection** sub-pane to choose which accounts or OUs to include or exclude. In single-account mode, this sub-pane does not appear in the console, so skip this step.

   In the **Account and organizational unit selection** pane, choose one of three methods for controlling administrative scope:
   - **Scope to entire AWS Organization (all OUs and accounts)** – No further definition is required.
   - **Scope to entire AWS Organization but exclude specific OUs and accounts** – Two sub-panes appear, one for **Organizational units** and one for **Accounts**. Choose **Add OUs** to open the **Add OUs** window, select the organizational units you need, and then choose **Add OUs**. To add individual accounts, in the **Accounts** sub-pane, choose **Add accounts**, select the accounts, and then choose **Add accounts**.
   - **Do not scope to entire AWS Organization, only include specific OUs and accounts** – Use the same sub-panes to add OUs or accounts.

3. Next, select resources in the **Resources** sub-pane. In the **Resource types** selection box, select all or specific AWS Shield Advanced resource types. For AWS Shield Advanced, the options are any or all of the following:
   - **Application Load Balancer**
   - **CloudFront Distribution**
   - **Classic Load Balancer**
   - **Elastic IP address**
4. For each selected resource type, choose one of the following mutually exclusive options:
   - **Include all resources**
   - **Include specific resources**
   - **Exclude specific resources**
5. If you choose **Include specific resources**, select one of the following expression types:
   - **Single criteria**
   - **Match all criteria (AND)**
   - **Match any criteria (OR)**
   - **NOT**
6. Each expression type supports single or multiple inclusion criteria and ARNs, so you can build combinations of tags or ARNs. All resource types support the **Tag** and **ARN** inclusion criteria. If you choose **Tag**, identify a **Key** and an optional **Value**. You can identify multiple tags.

> **CloudFront Region requirement**
>
> To protect **CloudFront Distribution** resources, use the US East (N. Virginia) Region (`us-east-1`).

To create a deployment (console)

A deployment binds your policies to a scope and activates enforcement. AWS Network Security Manager then applies your AWS Shield Advanced DDoS protections to the in-scope resources.

Complete the following steps:

1. On the **Overview** page, in the **Overview** pane, choose **Create deployment**.

2. Choose the policy that you want to deploy.
3. Choose the scope that defines the accounts and resources where AWS Network Security Manager applies the policy.
4. For **Deployment name**, enter a name. (Optional) For **Description**, enter a description.
5. (Optional) To make the deployment visible to other accounts in your organization, select **Enable cross-account visibility for this deployment**.
6. (Optional) To add tags, choose **Add new tag**, and then enter **Key** and optional **Value** combinations.
7. Choose **Create and deploy**. The **Deployments** page appears, where you can see the details of what AWS Network Security Manager created.

⚠ Deployments might incur charges

A deployment creates network security services and attaches them to a protected resource. AWS Network Security Manager charges for protected resources. The security services themselves protect the resource, which incurs additional charges. Before you deploy, review the AWS Network Security Manager pricing page and the AWS Shield Advanced pricing page.

AWS Network Security Manager and AWS Firewall Manager

AWS Network Security Manager is the service that is replacing AWS Firewall Manager. The two are wholly separate services, each with unique capabilities and a different pricing model. If you currently use AWS Firewall Manager, you don't need to take any action—your existing firewalls and policies continue to work. This chapter describes how the two services coexist, how precedence works when you use both, and how migration will work when it becomes available.

AWS Network Security Manager and AWS Firewall Manager operate side by side and independently of each other. At some point, AWS Firewall Manager will stop being sold and then stop operating.

Coexistence with AWS Firewall Manager

AWS Network Security Manager doesn't affect firewalls that you created with AWS Firewall Manager unless you initiate an action to change them. Firewalls and policies that you created with AWS Firewall Manager continue to operate unless you terminate them. This is true even after AWS Firewall Manager is retired.

Precedence during remediation

While both services operate, you can take actions that affect firewalls that you created with AWS Firewall Manager. When you create AWS Network Security Manager policies, you apply them to a scope, which is a set of target resources. If that scope includes resources that an AWS Firewall Manager policy already addresses, AWS Network Security Manager takes precedence during remediation. In that case, the AWS Network Security Manager policy and scope override the existing AWS Firewall Manager firewall and policy. The specific behavior depends on the resolution options that you select.

Planned migration from AWS Firewall Manager

Migrating from AWS Firewall Manager to AWS Network Security Manager consolidates your firewall management into a single service and lets you use its capabilities. Migrating is planned but is not yet available. When migration becomes available, you will be able to move your AWS Firewall Manager policies to AWS Network Security Manager. Migrating will require new actions in both services that are not yet in place.

The planned process will copy a selected policy from AWS Firewall Manager, transfer it to AWS Network Security Manager, and translate it into the new policy structure. This process doesn't affect your original policy or its associated firewalls in AWS Firewall Manager.

Security in AWS Network Security Manager

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

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

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

This documentation helps you understand how to apply the shared responsibility model when using AWS Network Security Manager. The following topics show you how to configure AWS Network Security Manager to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS Network Security Manager resources.

Topics

Data protection in AWS Network Security Manager

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

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

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with AWS Network Security Manager or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Encryption at rest

AWS Network Security Manager encrypts all data at rest using AWS Key Management Service (AWS KMS) keys. Service-managed KMS keys protect your resource configurations, deployment state, and synchronization data. You do not need to configure or manage these keys.

Encryption in transit

AWS Network Security Manager encrypts all data in transit using Transport Layer Security (TLS). All API requests require TLS 1.2 or later. Connections to the AWS Network Security Manager API endpoints use HTTPS exclusively.

Key management

AWS Network Security Manager uses service-managed AWS KMS keys for encryption at rest. You cannot provide your own KMS keys at this time.

Identity and access management for AWS Network Security Manager

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use AWS Network Security Manager resources. IAM is an AWS service that you can use with no additional charge.

Topics

Audience

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** - request permissions from your administrator if you cannot access features (see Troubleshooting AWS Network Security Manager identity and access)
- **Service administrator** - determine user access and submit permission requests (see How AWS Network Security Manager works with IAM)
- **IAM administrator** - write policies to manage access (see Identity-based policy examples for AWS Network Security Manager)

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

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

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How AWS Network Security Manager works with IAM

Before you use IAM to manage access to AWS Network Security Manager, learn what IAM features are available to use with AWS Network Security Manager.

IAM feature	AWS Network Security Manager support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys	Yes
ACLs	No
ABAC (tags in policies)	Yes
Temporary credentials	Yes
Principal permissions	Yes
Service roles	Yes
Service-linked roles	Yes

To get a high-level view of how AWS Network Security Manager and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for AWS Network Security Manager

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Resource-based policies within AWS Network Security Manager

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

AWS Network Security Manager does not support resource-based policies.

Policy actions for AWS Network Security Manager

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

To see a list of AWS Network Security Manager actions, see Actions Defined by AWS Network Security Manager in the *Service Authorization Reference*.

Policy actions in AWS Network Security Manager use the following prefix before the action:

```
network-security-manager
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "network-security-manager:action1",
      "network-security-manager:action2"
         ]
```

AWS Network Security Manager defines IAM actions across its resource types. These include create, read, update, and delete (CRUD) operations for rules, templates, policies, scopes, and deployments, along with snapshot, tagging, administrative, synchronization status, and resource association operations.

Policy resources for AWS Network Security Manager

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of AWS Network Security Manager resource types and their ARNs, see Resources Defined by AWS Network Security Manager in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by AWS Network Security Manager.

AWS Network Security Manager defines an IAM resource type for each of its resources (rule, template, policy, scope, and deployment), plus a snapshot type for each one (for example, `rule-snapshot`). The ARN format is:

```
arn:aws:network-security-manager:{region}:{account}:{resource-type}:{resource-id}
```

Draft ARNs append `:DRAFT` and snapshot ARNs append the version number (for example, `:3`).

Drafts are not separate IAM resource types. For authorization, a draft is treated as a child of its active resource, which is the parent. A `:DRAFT`-qualified ARN authorizes against the active resource, so permission to act on a draft comes from permission to act on that active resource.

Snapshots differ from drafts. Each snapshot is its own IAM resource type, so a version-qualified ARN authorizes against that snapshot resource type directly, not against the parent active resource.

Policy condition keys for AWS Network Security Manager

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element specifies when statements execute based on defined criteria. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of AWS Network Security Manager condition keys, see Condition Keys for AWS Network Security Manager in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions Defined by AWS Network Security Manager.

AWS Network Security Manager supports the following AWS global tag condition keys for attribute-based access control (ABAC):

- `aws:ResourceTag/${TagKey}`
- `aws:RequestTag/${TagKey}`
- `aws:TagKeys`

Tag-based condition keys apply to active resources and snapshots. Draft resources are not taggable and do not support `aws:ResourceTag` conditions.

ACLs in AWS Network Security Manager

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

AWS Network Security Manager does not support access control lists (ACLs).

ABAC with AWS Network Security Manager

Supports ABAC (tags in policies): Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

AWS Network Security Manager supports attribute-based access control using all three AWS global tag condition keys. You can restrict access to AWS Network Security Manager resources based on the tags attached to those resources.

Using temporary credentials with AWS Network Security Manager

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM and AWS services that work with IAM in the *IAM User Guide*.

Cross-service principal permissions for AWS Network Security Manager

Supports forward access sessions (FAS): Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see Forward access sessions.

Service roles for AWS Network Security Manager

Supports service roles: Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
>
> Changing the permissions for a service role might break AWS Network Security Manager functionality. Edit service roles only when AWS Network Security Manager provides guidance to do so.

Service-linked roles for AWS Network Security Manager

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

AWS Network Security Manager uses a service-linked role to operate in member accounts of your organization. AWS Network Security Manager creates the service-linked role automatically when you enable the service. The role grants AWS Network Security Manager permission to manage AWS WAF and AWS Shield Advanced resources. The role also grants access to AWS Config for resource discovery and permission to perform cross-account operations.

For details about creating and managing AWS Network Security Manager service-linked roles, see the section called “Service-linked roles”.

Identity-based policy examples for AWS Network Security Manager

By default, users and roles don't have permission to create or modify AWS Network Security Manager resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by AWS Network Security Manager, including the format of the ARNs for each of the resource types, see Actions, Resources, and Condition Keys for AWS Network Security Manager in the *Service Authorization Reference*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS Network Security Manager resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies

adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.

- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the AWS Network Security Manager console

To access the AWS Network Security Manager console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the AWS Network Security Manager resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To access the AWS Network Security Manager console, a principal needs at minimum permissions for the `network-security-manager:List*` and `network-security-manager:Get*` actions.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

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

```
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Read-only access to AWS Network Security Manager

The following policy grants a principal read-only access to all AWS Network Security Manager resources. This is useful for auditors or monitoring roles that need to view configurations and synchronization status without making changes.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "network-security-manager:Get*",
                "network-security-manager:List*"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

Manage rules and templates only

The following policy allows a principal to create, update, and delete rules and templates, without granting access to policies, scopes, or deployments. This is useful for teams that author security configurations but do not control enforcement.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "network-security-manager:CreateRule",
                "network-security-manager:GetRule",
                "network-security-manager:UpdateRule",
                "network-security-manager:DeleteRule",
                "network-security-manager:ListRules",
                "network-security-manager:CreateRuleSnapshot",
                "network-security-manager:ListRuleSnapshots",
                "network-security-manager:CreateTemplate",
                "network-security-manager:GetTemplate",
                "network-security-manager:UpdateTemplate",
                "network-security-manager:DeleteTemplate",
                "network-security-manager:ListTemplates",
                "network-security-manager:CreateTemplateSnapshot",
                "network-security-manager:ListTemplateSnapshots",
                "network-security-manager:TagResource",
                "network-security-manager:UntagResource",
                "network-security-manager:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Troubleshooting AWS Network Security Manager identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with AWS Network Security Manager and IAM.

Topics

I am not authorized to perform an action in AWS Network Security Manager

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `network-security-manager:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 network-security-manager:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `network-security-manager:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to AWS Network Security Manager.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in AWS Network Security Manager. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my AWS Network Security Manager resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether AWS Network Security Manager supports these features, see How AWS Network Security Manager works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Using service-linked roles for AWS Network Security Manager

AWS Network Security Manager uses IAM service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to AWS Network Security Manager. Service-linked roles are predefined by AWS Network Security Manager and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up AWS Network Security Manager easier because you don't have to manually add the necessary permissions. AWS Network Security Manager defines the

permissions of its service-linked roles. Unless defined otherwise, only AWS Network Security Manager can assume its roles. The defined permissions include the trust policy and the permissions policy. You can't attach that permissions policy to any other IAM entity.

You can delete a service-linked role only after you first delete the related resources. This protects your AWS Network Security Manager resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS services that work with IAM and look for the services that include **Yes** in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Service-linked role permissions for AWS Network Security Manager

AWS Network Security Manager uses the service-linked role named `AWSServiceRoleForNetworkSecurityManager`. This service-linked role trusts the `network-security-manager.amazonaws.com` service principal to assume the role.

The role permissions policy named `NetworkSecurityManagerServiceRolePolicy` is an AWS managed policy that allows AWS Network Security Manager to complete actions on the specified resources.

The following trust policy allows the `network-security-manager.amazonaws.com` service principal to assume the role.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "network-security-manager.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

> **Note**
>
> You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

Permissions details

AWS Network Security Manager requires each group of permissions to protect and discover resources across your account and organization. The `NetworkSecurityManagerServiceRolePolicy` managed policy grants the following permissions, grouped by service:

- `wafv2` – Create, read, update, and delete web ACLs and Firewall Manager rule groups. Associate and disassociate web ACLs with protected resources. Manage web ACL logging configurations and rule group permission policies. Tag and untag resources. Validate AWS Network Security Manager web ACL and rule configurations.
- `shield` – Create, describe, list, and delete AWS Shield Advanced protections. Create and describe the AWS Shield Advanced subscription.
- `elasticloadbalancing` – Describe Application Load Balancers and tags. Create, delete, and set web ACL associations on Application Load Balancers.
- `cloudfront` – Get and list Amazon CloudFront distributions, configurations, and tags. Associate and disassociate distribution web ACLs. Read VPC origins.
- `apigateway` – Read Amazon API Gateway REST APIs, stages, and tags. Set the web ACL on an API stage.
- `ec2` – Read Amazon EC2 Verified Access instance web ACLs and associations. Describe Elastic IP addresses for AWS Shield Advanced protection.
- `cognito-idp`, `apprunner`, `appsync`, `amplify`, and `bedrock-agentcore` – Read web ACL associations for Amazon Cognito user pools, AWS App Runner services, AWS AppSync GraphQL APIs, AWS Amplify apps, and Amazon Bedrock AgentCore gateways so AWS Network Security Manager can discover protected resources.
- `config` – Select resource configurations and manage the service-linked AWS Config configuration recorder that AWS Network Security Manager uses to discover in-scope resources.
- `organizations` – Read your organization, accounts, roots, organizational units, and delegated administrators so AWS Network Security Manager can operate across your organization.

- `logs` – Create, read, update, delete, and list log deliveries for web ACL logging.
- `s3` – Get and put bucket policies on Amazon S3 buckets whose names begin with `aws-waf-logs-` so web ACL logs can be delivered.
- `iam` – Create the service-linked roles that AWS Shield Advanced, AWS Config, and AWS WAF require, scoped by the `iam:AWSServiceName` condition key.

The following JSON shows the complete `NetworkSecurityManagerServiceRolePolicy` permissions policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LogsGeneral",
      "Effect": "Allow",
      "Action": [
        "logs:ListLogDeliveries",
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs:DeleteLogDelivery"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3LogBucketGeneral",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketPolicy",
        "s3:PutBucketPolicy"
      ],
      "Resource": "arn:aws:s3:::aws-waf-logs-*"
    },
    {
      "Sid": "CloudfrontVpcOriginAccess",
      "Effect": "Allow",
      "Action": [
        "cloudfront:GetVpcOrigin"
      ],
      "Resource": "arn:aws:cloudfront::*:vpcorigin/*"
    },
```

```
    {
      "Sid": "ConfigUnscoped",
      "Effect": "Allow",
      "Action": [
        "config:SelectResourceConfig"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DescribeOrganizationUnScoped",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeOrganization",
        "organizations:ListAccounts",
        "organizations:ListChildren",
        "organizations:ListRoots",
        "organizations:ListParents",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListAWSServiceAccessForOrganization"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DescribeAccountScoped",
      "Effect": "Allow",
      "Action": "organizations:DescribeAccount",
      "Resource": "arn:aws:organizations::*:account/o-*/*"
    },
    {
      "Sid": "DescribeOrganizationUnitScoped",
      "Effect": "Allow",
      "Action": "organizations:DescribeOrganizationalUnit",
      "Resource": "arn:aws:organizations::*:ou/o-*/ou-*"
    },
    {
      "Sid": "ListDelegatedAdministratorsScoped",
      "Effect": "Allow",
      "Action": "organizations:ListDelegatedAdministrators",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "organizations:ServicePrincipal": [
            "network-security-manager.amazonaws.com"
          ]
```

```
        }
      }
    },
    {
      "Sid": "LoadBalancerNonMutationApiAccess",
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTags",
        "elasticloadbalancing:DescribeWebACLAssociation"
      ],
      "Resource": "*"
    },
    {
      "Sid": "LoadBalancerGetWebACLScoped",
      "Effect": "Allow",
      "Action": "elasticloadbalancing:GetLoadBalancerWebACL",
      "Resource": "arn:aws:elasticloadbalancing:*:*:loadbalancer/app/*/*"
    },
    {
      "Sid": "CloudFrontDistributionNonMutationApiAccess",
      "Effect": "Allow",
      "Action": [
        "cloudfront:GetDistribution",
        "cloudfront:GetDistributionConfig",
        "cloudfront:ListTagsForResource"
      ],
      "Resource": "arn:aws:cloudfront::*:distribution/*"
    },
    {
      "Sid": "ApiGatewayStageNonMutationApiAccess",
      "Effect": "Allow",
      "Action": "apigateway:GET",
      "Resource": [
        "arn:aws:apigateway:*::/restapis",
        "arn:aws:apigateway:*::/restapis/*/stages",
        "arn:aws:apigateway:*::/restapis/*/stages/*",
        "arn:aws:apigateway:*::/tags/*"
      ]
    },
    {
      "Sid": "Wafv2General",
      "Effect": "Allow",
      "Action": [
```

```
        "wafv2:TagResource",
        "wafv2:ListResourcesForWebACL",
        "wafv2:AssociateWebACL",
        "wafv2:ListTagsForResource",
        "wafv2:UntagResource",
        "wafv2:GetWebACL",
        "wafv2:DisassociateFirewallManager",
        "wafv2:DeleteWebACL",
        "wafv2:DisassociateWebACL",
        "wafv2:ValidateNetworkSecurityManagerWebACLConfiguration",
        "wafv2:ValidateNetworkSecurityManagerRuleConfiguration"
      ],
      "Resource": "arn:aws:wafv2:*:*:*/webacl/*"
    },
    {
      "Sid": "Wafv2WebAclAndRuleGroupMutation",
      "Effect": "Allow",
      "Action": [
        "wafv2:UpdateWebACL",
        "wafv2:CreateWebACL",
        "wafv2:DeleteFirewallManagerRuleGroups",
        "wafv2:PutFirewallManagerRuleGroups"
      ],
      "Resource": [
        "arn:aws:wafv2:*:*:*/webacl/*",
        "arn:aws:wafv2:*:*:global/rulegroup/*",
        "arn:aws:wafv2:*:*:regional/rulegroup/*",
        "arn:aws:wafv2:*:*:global/managedruleset/*",
        "arn:aws:wafv2:*:*:regional/managedruleset/*",
        "arn:aws:wafv2:*:*:global/ipset/*",
        "arn:aws:wafv2:*:*:regional/ipset/*",
        "arn:aws:wafv2:*:*:global/regexpatternset/*",
        "arn:aws:wafv2:*:*:regional/regexpatternset/*"
      ]
    },
    {
      "Sid": "Wafv2Logging",
      "Effect": "Allow",
      "Action": [
        "wafv2:PutLoggingConfiguration",
        "wafv2:GetLoggingConfiguration",
        "wafv2:DeleteLoggingConfiguration"
      ],
      "Resource": "arn:aws:wafv2:*:*:*/webacl/*"
```

```
    },
    {
      "Sid": "Wafv2ListLogging",
      "Effect": "Allow",
      "Action": "wafv2:ListLoggingConfigurations",
      "Resource": "*"
    },
    {
      "Sid": "Wafv2ListWebACLs",
      "Effect": "Allow",
      "Action": "wafv2:ListWebACLs",
      "Resource": "*"
    },
    {
      "Sid": "Wafv2PermissionPolicy",
      "Effect": "Allow",
      "Action": [
        "wafv2:PutPermissionPolicy",
        "wafv2:GetPermissionPolicy",
        "wafv2:DeletePermissionPolicy"
      ],
      "Resource": [
        "arn:aws:wafv2:*:*:global/rulegroup/*",
        "arn:aws:wafv2:*:*:regional/rulegroup/*"
      ]
    },
    {
      "Sid": "Wafv2WebaclDescribe",
      "Effect": "Allow",
      "Action": "wafv2:GetWebACLForResource",
      "Resource": "arn:aws:wafv2:*:*:*/webacl/*"
    },
    {
      "Sid": "CloudFrontListDistributions",
      "Effect": "Allow",
      "Action": [
        "cloudfront:ListDistributionsByWebACLId",
        "cloudfront:ListDistributions"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CloudFrontDistributionMutation",
      "Effect": "Allow",
```

```
      "Action": [
        "cloudfront:AssociateDistributionWebACL",
        "cloudfront:DisassociateDistributionWebACL"
      ],
      "Resource": "arn:aws:cloudfront::*:distribution/*"
    },
    {
      "Sid": "ElbWebAclRemediationScoped",
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:CreateWebACLAssociation",
        "elasticloadbalancing:DeleteWebACLAssociation"
      ],
      "Resource": "arn:aws:elasticloadbalancing:*:*:loadbalancer/app/*/*"
    },
    {
      "Sid": "ElbSetWebAcl",
      "Effect": "Allow",
      "Action": "elasticloadbalancing:SetWebACL",
      "Resource": "*"
    },
    {
      "Sid": "ApiGatewayWebAclRemediation",
      "Effect": "Allow",
      "Action": "apigateway:SetWebACL",
      "Resource": "arn:aws:apigateway:*::/restapis/*/stages/*"
    },
    {
      "Sid": "CognitoWebAclIntegration",
      "Effect": "Allow",
      "Action": [
        "cognito-idp:GetWebACLForResource",
        "cognito-idp:ListResourcesForWebACL"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AppRunnerWebAclIntegration",
      "Effect": "Allow",
      "Action": [
        "apprunner:DescribeWebAclForService",
        "apprunner:ListAssociatedServicesForWebAcl"
      ],
      "Resource": "*"
```

```
    },
    {
      "Sid": "VerifiedAccessWebAclIntegration",
      "Effect": "Allow",
      "Action": [
        "ec2:GetVerifiedAccessInstanceWebAcl",
        "ec2:DescribeVerifiedAccessInstanceWebAclAssociations"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AppSyncWebAclIntegration",
      "Effect": "Allow",
      "Action": [
        "appsync:GetWebACLForResource",
        "appsync:ListResourcesForWebACL"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AmplifyWebAclIntegration",
      "Effect": "Allow",
      "Action": [
        "amplify:GetWebACLForResource",
        "amplify:ListResourcesForWebACL"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ShieldGeneral",
      "Effect": "Allow",
      "Action": [
        "shield:CreateProtection",
        "shield:DeleteProtection",
        "shield:DescribeProtection",
        "shield:ListProtections",
        "shield:CreateSubscription",
        "shield:DescribeSubscription",
        "shield:GetSubscriptionState",
        "ec2:DescribeAddresses",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTags"
      ],
      "Resource": "*"
```

```
    },
    {
      "Sid": "ShieldSlrCreation",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/shield.amazonaws.com/*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "shield.amazonaws.com"
        }
      }
    },
    {
      "Sid": "AllowConfigRecorderList",
      "Effect": "Allow",
      "Action": "config:ListConfigurationRecorders",
      "Resource": "*"
    },
    {
      "Sid": "AllowPutServiceLinkedConfigRecorder",
      "Effect": "Allow",
      "Action": "config:PutServiceLinkedConfigurationRecorder",
      "Resource": "*"
    },
    {
      "Sid": "AllowConfigRecorderScopedAccess",
      "Effect": "Allow",
      "Action": [
        "config:DescribeConfigurationRecorders",
        "config:DescribeConfigurationRecorderStatus",
        "config:DeleteServiceLinkedConfigurationRecorder",
        "config:AssociateResourceTypes",
        "config:DisassociateResourceTypes"
      ],
      "Resource": "arn:aws:config:*:*:configuration-recorder/*"
    },
    {
      "Sid": "AllowCreateConfigServiceLinkedRole",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/config.amazonaws.com/*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "config.amazonaws.com"
```

```
        }
      }
    },
    {
      "Sid": "AllowListActionsForAgentCoreGateway",
      "Effect": "Allow",
      "Action": [
        "bedrock-agentcore:GatewayListResourcesForWebACL"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowGetActionForAgentCoreGateway",
      "Effect": "Allow",
      "Action": [
        "bedrock-agentcore:GatewayGetWebACLForResource"
      ],
      "Resource": "arn:aws:bedrock-agentcore:*:*:gateway/*"
    },
    {
      "Sid": "WafV2LoggingSlrCreation",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/wafv2.amazonaws.com/*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "wafv2.amazonaws.com"
        }
      }
    }
  ]
}
```

Creating a service-linked role for AWS Network Security Manager

You don't need to manually create a service-linked role. When you enable AWS Network Security Manager, AWS Network Security Manager creates the service-linked role for you. There is no manual step to perform.

If you delete this service-linked role and then need to create it again, AWS Network Security Manager recreates the role the next time you use the service. You don't have to take any action.

For more information, see Creating a service-linked role in the *IAM User Guide*.

Editing a service-linked role for AWS Network Security Manager

You can't change the permissions of the `AWSServiceRoleForNetworkSecurityManager` service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM.

For more information, see Editing a service-linked role in the *IAM User Guide*.

Deleting a service-linked role for AWS Network Security Manager

If you no longer need to use AWS Network Security Manager, we recommend that you delete the `AWSServiceRoleForNetworkSecurityManager` service-linked role.

Before you delete the service-linked role, clean up the resources that AWS Network Security Manager manages. Disable AWS Network Security Manager and remove the policies and scopes so that no resources that AWS Network Security Manager manages remain in your account. You can then delete the role manually.

> **Note**
>
> If AWS Network Security Manager is using the role when you try to delete the resources, the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To manually delete the service-linked role using IAM

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**.
3. In the search box, enter `AWSServiceRoleForNetworkSecurityManager`, and then select the role.
4. Choose **Delete**.
5. To confirm the deletion, enter the role name, and then choose **Delete**.

For more information, see Deleting a service-linked role in the *IAM User Guide*.

Supported Regions for AWS Network Security Manager service-linked roles

AWS Network Security Manager supports using service-linked roles in all of the AWS Regions where AWS Network Security Manager is available.

Compliance validation for AWS Network Security Manager

Our new AWS sign-up experience is not designed for regulated workloads. If you're using our new AWS sign-up experience, but you want to use AWS for regulated workloads, you can sign up for AWS (advanced) or activate advanced features for your AWS environment.

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

Resilience in AWS Network Security Manager

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

In addition to the AWS global infrastructure, AWS Network Security Manager offers several features to help support your data resiliency and backup needs.

Infrastructure Security in AWS Network Security Manager

As a managed service, AWS Network Security Manager is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access AWS Network Security Manager through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Security best practices for AWS Network Security Manager

The following best practices help you maintain a secure AWS Network Security Manager configuration.

Preventative controls

- **Use IAM least privilege** -- Grant only the permissions required for each user or role to perform their AWS Network Security Manager tasks. Use IAM condition keys to restrict access to specific resources when possible.
- **Use service-linked roles** -- AWS Network Security Manager uses service-linked roles to perform actions on your behalf. These roles have pre-defined permissions scoped to the minimum required access.
- **Configure scopes carefully** -- Define scope configurations that target only the accounts and resource types that require protection. Overly broad scopes can lead to unintended enforcement.

Detective controls

- **Enable AWS CloudTrail logging** -- CloudTrail records all AWS Network Security Manager API calls. Review these logs to audit who made changes to your security configurations and when.

- **Monitor synchronization status** -- Regularly check the synchronization status of your deployments to identify resources that are out of sync with your intended security posture.

Monitoring AWS Network Security Manager

Monitoring AWS Network Security Manager helps you understand whether your resources are protected as intended. AWS Network Security Manager provides the following monitoring capabilities:

- **Synchronization status APIs** -- Query the protection state of individual resources across your deployments.
- **AWS CloudTrail** -- Record all AWS Network Security Manager API calls for auditing, compliance, and operational troubleshooting.

Monitoring synchronization status

The primary way to monitor whether your resources are protected is through the synchronization status APIs. AWS Network Security Manager evaluates each in-scope resource and records whether its actual firewall configuration matches the intended configuration.

Synchronization status operations

AWS Network Security Manager provides two operations for querying synchronization status:

ListResourceSynchronizationStatuses

Returns per-resource status for a specific deployment. Requires a `deploymentIdentifier` parameter. Optionally filter by `synchronizationStatus` (IN_SYNC, OUT_OF_SYNC, or NOT_APPLICABLE).

ListAggregateResourceSynchronizationStatuses

Returns per-resource aggregate status across all of your deployments. Optionally filter by `synchronizationStatus`. This operation shows the combined result of all policies that apply to each resource.

Both operations support pagination with `maxResults` (1 to 100) and `nextToken`.

Understanding synchronization status values

IN_SYNC

The resource's actual firewall configuration matches the effective configuration computed from all applicable policies.

OUT_OF_SYNC

The resource's actual configuration differs from the effective configuration. The `outOfSyncReasons` field provides per-setting details showing expected versus actual values.

NOT_APPLICABLE

The resource's account and resource type are in scope, but the resource itself does not match the scope filters (for example, a tag filter excludes it). No action is needed.

If a resource does not appear in the results at all, the account or resource type is not in scope for the deployment. Check your scope configuration.

Understanding evaluation timestamps

Each synchronization status entry includes two timestamps:

- `evaluatedAt` -- When AWS Network Security Manager last evaluated the resource's configuration. A stale `evaluatedAt` value indicates that evaluation has not yet run for recent changes.
- `updatedAt` -- When the synchronization status record itself last changed. If `updatedAt` is older than `evaluatedAt`, the status has been stable since the last evaluation.

Cross-account visibility

When multiple administrators protect the same resource, visibility into the aggregate synchronization status depends on the `enableCrossAccountVisibility` setting on each deployment.

- If all deployments covering a resource have cross-account visibility enabled, any administrator can see the full aggregate status and out-of-sync reasons.

- If any deployment has cross-account visibility disabled, other administrators see a `NOT_VISIBLE` marker instead of detailed reasons. This indicates that another administrator also protects the resource but has not shared visibility.

Diagnosing out-of-sync resources

When a resource reports OUT_OF_SYNC, the `outOfSyncReasons` field identifies the specific configuration differences. The top-level reason is one of:

- `missingFirewall` -- No firewall protection is associated with the resource.
- `invalidFirewall` -- A firewall exists but its configuration does not match expectations. Specific differences are listed in categorized buckets.

If remediation is enabled but cannot fix the resource, the `remediationIssues` field provides a `correctiveAction` with guidance on how to resolve the issue manually. Remediation is a policy-level setting that is turned off by default. For more information, see the section called "Synchronization and remediation".

Multiple deployments with different configurations can target the same resource. In this case, the per-deployment synchronization status can be OUT_OF_SYNC while the aggregate synchronization status is IN_SYNC. This is expected behavior, not a failure. A higher-priority policy's settings took precedence during merging. As a result, the resource matches the effective configuration, even though it does not match the configuration of the lower-priority deployment.

Logging AWS Network Security Manager API calls with AWS CloudTrail

AWS Network Security Manager is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all API calls for AWS Network Security Manager as management events. The calls captured include calls from the AWS Network Security Manager console and programmatic calls to the AWS Network Security Manager API operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Network Security Manager. If you do not configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. For more information, see the AWS CloudTrail User Guide.

AWS Network Security Manager information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS Network Security Manager, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail Event history.

For an ongoing record of events in your AWS account, including events for AWS Network Security Manager, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition. It delivers the log files to the Amazon S3 bucket that you specify. You can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail

All AWS Network Security Manager operations are logged by CloudTrail. The public operations include the following:

- Create, read, update, and delete (CRUD) operations for rules, templates, policies, scopes, and deployments
- Snapshot operations
- Tagging operations
- Administrator account operations
- Synchronization status operations
- Resource association operations
- Rule generation operations

Every event entry includes the following information:

- The event source: `network-security-manager.amazonaws.com`
- The event name, which is the API operation name (for example, `CreatePolicy` or `ListDeployments`)

- The resources affected, identified by ARN and resource type

Read-only operations (such as `GetPolicy`, `ListRules`, and `ListResourceSynchronizationStatuses`) are logged with `readOnly: true` and do not include response elements. Mutating operations include request parameters and response elements.

> **Note**
>
> CloudTrail redacts sensitive fields from event records. For example, CloudTrail redacts the `prompt` parameter in the `GenerateRuleConfiguration` operation.

Example: CreatePolicy event

The following example shows a CloudTrail log entry for the `CreatePolicy` operation.


```
{
  "eventVersion": "1.09",
  "userIdentity": {
      "type": "AssumedRole",
      "principalId": "AROAEXAMPLEID:my-session",
      "arn": "arn:aws:sts::123456789012:assumed-role/AdminRole/my-session",
      "accountId": "123456789012",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "sessionContext": {
          "sessionIssuer": {
              "type": "Role",
              "principalId": "AROAEXAMPLEID",
              "arn": "arn:aws:iam::123456789012:role/AdminRole",
              "accountId": "123456789012",
              "userName": "AdminRole"
          },
          "attributes": {
              "creationDate": "2026-08-10T18:22:41Z",
              "mfaAuthenticated": "false"
          }
      }
  },
  "eventTime": "2026-08-10T18:24:03Z",
  "eventSource": "network-security-manager.amazonaws.com",
```

```
  "eventName": "CreatePolicy",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.44",
  "userAgent": "aws-cli/2.17.0 Python/3.12.4 Darwin/24.6.0",
  "requestParameters": {
      "clientToken": "550e8400-e29b-41d4-a716-446655440001",
      "policyName": "production-web-acl-policy",
      "firewallType": "WAF",
      "priority": 1,
      "associatedTemplateAndRuleList": [
          {
              "templateIdentifier": "arn:aws:network-security-manager:us-
east-1:123456789012:template:t1a2b3c4d5"
          }
      ],
      "policyConfiguration": {
          "remediationEnabled": true,
          "resourcesCleanUp": false,
          "wafConfig": {
              "existingCustomerWebACLResolution": "RETROFIT",
              "conflictResolution": "MERGE_WHERE_APPLICABLE"
          }
      },
      "isPublished": true
  },
  "responseElements": {
      "policyId": "p9z8y7x6w5v4u3t2",
      "policyArn": "arn:aws:network-security-manager:us-
east-1:123456789012:policy:p9z8y7x6w5v4u3t2",
      "policyName": "production-web-acl-policy",
      "status": "ACTIVE",
      "version": "1"
  },
  "requestID": "b1c2d3e4-5f67-4890-a1b2-c3d4e5f6a7b8",
  "eventID": "f0e1d2c3-b4a5-4968-8776-5a4b3c2d1e0f",
  "readOnly": false,
  "resources": [
      {
          "accountId": "123456789012",
          "type": "AWS::NetworkSecurityManager::Policy",
          "ARN": "arn:aws:network-security-manager:us-
east-1:123456789012:policy:p9z8y7x6w5v4u3t2"
      }
  ],
```

```
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

AWS Network Security Manager quotas

Your AWS account has the following default quotas for AWS Network Security Manager. Unless otherwise noted, each quota is per account, per Region.

Resource quotas

The following table lists the maximum number of AWS Network Security Manager resources you can create per account in each Region.

Resource	Default quota	Adjustable
Rules per account	200	No
Templates per account	100	No
Policies per account	100	No
Scopes per account	100	No
Deployments per account	100	No
Snapshots per resource	10	No

Resource composition quotas

The following table lists the quotas for how AWS Network Security Manager resources reference each other within the resource hierarchy.

Quota	Value
Rules per template	1 to 50
Templates per AWS WAF policy	1 to 2
Rules and templates combined per AWS WAF policy	1 to 100

Quota	Value
Templates or rules per AWS Shield Advanced policy	0 (must be empty)
Policies per deployment	1 to 2
Scopes per deployment	1 (exactly one)

Scope configuration quotas

The following table lists the quotas for scope configuration expressions.

Quota	Value
Resource type scopes per scope	20
Scope expression depth (nested levels)	2
Scope expression breadth (criteria per level)	20

Administrator account quotas

The following table lists the quotas for AWS Network Security Manager administrator account priority levels.

Quota	Value
Administrator priority levels	1 to 10

API pagination quotas

The following table lists the pagination limits for List operations.

Quota	Value
Maximum results per page (`maxResults`)	100
Minimum results per page (`maxResults`)	1

Quota adjustability

The quotas listed in the preceding sections are fixed and cannot be increased at this time.

Document history for the AWS Network Security Manager Developer Guide

The following table describes the documentation releases for AWS Network Security Manager.

Change	Description	Date
Initial release	Initial release of the AWS Network Security Manager Developer Guide.	August 1, 2026