



Network Access Analyzer

Amazon Virtual Private Cloud



Amazon Virtual Private Cloud: Network Access Analyzer

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

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

This documentation is a draft for private preview for regions in the AWS European Sovereign Cloud. Documentation content will continue to evolve. Published: January 13, 2026.

Table of Contents

What is Network Access Analyzer	1
Concepts	2
Access Network Access Analyzer	2
Pricing	3
How Network Access Analyzer works	4
Supported source and destination resources	4
Supported path resources	5
Unsupported resources	6
Unsupported network configurations	6
Limitations	7
Getting started	9
Analyze your network	9
Review your findings	10
Delete a Network Access Scope	11
Getting started using the CLI	12
Step 1: Create a Network Access Scope	12
Step 2: Analyze a Network Access Scope	16
Step 3: Get the results of a Network Access Scope analysis	16
Network Access Scopes	22
Resource statements	22
Packet header statements	23
Match conditions	24
Exclusion conditions	25
Example Network Access Scopes	25
Identity and access management	32
Audience	32
Authenticating with identities	33
AWS account root user	33
Federated identity	33
IAM users and groups	33
IAM roles	34
Managing access using policies	34
Identity-based policies	34
Resource-based policies	35

Other policy types	35
Multiple policy types	35
How Network Access Analyzer works with IAM	35
Identity-based policies	36
Resource-based policies	37
Policy actions	37
Policy resources	38
Policy condition keys	39
ACLs	39
ABAC	39
Temporary credentials	40
Principal permissions	40
Service roles	40
Service-linked roles	40
Required API permissions	40
Additional information	41
AWS managed policies	42
AmazonVPCNetworkAccessAnalyzerFullAccessPolicy	43
Policy updates	43
Quotas	45
Analysis runtime	45
Troubleshooting	46
Document history	47

What is Network Access Analyzer?

Network Access Analyzer is a feature that identifies unintended network access to your resources on AWS. You can use Network Access Analyzer to specify your network access requirements and to identify potential network paths that do not meet your specified requirements. You can use Network Access Analyzer to:

- **Understand, verify, and improve your network security posture** – Network Access Analyzer helps you identify unintended network access relative to your security and compliance requirements, enabling you to take steps to improve your network security.
- **Demonstrate compliance** – Network Access Analyzer helps you demonstrate that your network on AWS meets your compliance requirements.

Network Access Analyzer can help you verify the following example requirements:

- **Network segmentation** – Verify that your production environment VPCs and development environment VPCs are isolated from one another. Likewise, you can verify that a separate logical network is used for systems that process credit card information, and that it's isolated from the rest of your environment.
- **Internet accessibility** – Identify resources in your environment that can be accessed from internet gateways, and verify that they are limited to only those resources that have a legitimate need to be accessible from the internet.
- **Trusted network paths** – Verify that you have appropriate network controls such as network firewalls and NAT gateways on all network paths between your resources and internet gateways.
- **Trusted network access** – Verify that your resources have network access only from a trusted IP address range, over specific ports and protocols. You can specify your network access requirements in terms of:
 - Resource IDs (for example, vpc-1234567890abcdef0)
 - Resource types (for example, AWS::EC2::InternetGateway)
 - Resource tags
 - IP address ranges, port ranges, and traffic protocols

Network Access Analyzer concepts

The following are the key concepts for Network Access Analyzer:

Network Access Scopes

You can specify your network access requirements as Network Access Scopes, which determine the types of findings that the analysis produces. You add entries to **MatchPaths** to specify the types of network paths to identify. You add entries to **ExcludePaths** to specify the types of network paths to exclude.

- **MatchPaths** – Specifies the types of network paths that an analysis produces. Typically, you specify network paths that you consider to be a violation of your security or compliance requirements. For example, if you don't want to allow network paths that start in VPC A and end in VPC B, specify VPC A as a source and VPC B as a destination. When you analyze this Network Access Scope, you would see any findings that indicate any potential network paths that start in VPC A and end in VPC B.
- **ExcludePaths** – Prevents certain network paths from appearing in your findings. Typically, you specify network paths that you consider to be a legitimate exception to your network security or compliance requirements. For example, to identify all network interfaces that are reachable from an internet gateway except for your web servers, specify the relevant paths using **MatchPaths**, and then exclude any path with your web servers as a destination using **ExcludePaths**. When you analyze this Network Access Scope, you would see any network paths that originate from an internet gateway and end at a network interface, except for any paths that end at your web servers.

Findings

Findings are potential paths in your network that match any of the **MatchPaths** entries in your Network Access Scope, but do not match any of the **ExcludePaths** entries in your Network Access Scope.

Access Network Access Analyzer

You can use any of the following interfaces to access and work with Network Access Analyzer:

- **AWS Management Console** – Provides a web interface that you can use to create and manage Network Access Analyzer resources.

- **AWS Command Line Interface (AWS CLI)** – Provides commands for AWS services including Network Access Analyzer. The AWS CLI is supported on Windows, macOS, and Linux. For more information, see the [AWS Command Line Interface User Guide](#).
- **CloudFormation** – Create templates to provision and manage AWS resources as a single unit. For more information, see [AWS::EC2::NetworkInsightsAccessScope](#) and [AWS::EC2::NetworkInsightsAccessScopeAnalysis](#).
- **AWS SDKs** – Provides language-specific APIs and takes care of many of the connection details, such as calculating signatures, and handling request retries and errors. For more information, see [Tools to build on AWS](#).
- **Query API** – Provides low-level API actions that you call using HTTPS requests. Using the Query API is the most direct way to access Network Access Analyzer. However, the Query API requires your application to handle low-level details such as generating the hash to sign the request and handling errors. For more information, see [Amazon VPC actions](#) in the *Amazon EC2 API Reference*.

Pricing

When you run a Network Access Analyzer analysis, you are charged based on the number of network interfaces that are analyzed. For more information, see [Pricing](#).

How Network Access Analyzer works

Network Access Analyzer uses automated reasoning algorithms to analyze the network paths that a packet can take between resources in an AWS network. It then produces findings for paths that match a customer defined Network Access Scope. Network Access Analyzer performs a static analysis of a network configuration, meaning that no packets are transmitted in the network as part of this analysis. Because Network Access Analyzer only considers the state of the network as described in the network configuration, packet loss that is due to transient network interruptions or service failures is not considered in the analysis.

Network Access Analyzer makes a best effort attempt to return a diverse, representative set of findings from among all possible findings.

Not all AWS network configurations are supported by Network Access Analyzer. The following sections describe the types of network paths that Network Access Analyzer produces as findings. For more information about resources that you can reference in Network Access Scopes, see [Network Access Scopes](#).

Contents

- [Supported source and destination resources](#)
- [Supported path resources](#)
- [Unsupported resources](#)
- [Unsupported network configurations](#)
- [Limitations](#)

Supported source and destination resources

A Network Access Analyzer finding is a network path that a packet can take in a network. Network Access Analyzer can only produce findings for network paths that start or end at the following types of resources:

- Internet gateways
- Network interfaces
- Transit gateway attachments

- VPC interface endpoints
- VPC gateway endpoints
- VPC gateway load balancer endpoints
- VPC service endpoints
- VPC peering connections
- Virtual private gateways

Supported path resources

A Network Access Analyzer network path can pass through multiple resources from the start to the end of the network path. Only the following resource types are supported as resources on network paths in Network Access Analyzer findings:

- Internet gateways
- Load balancers
- NAT gateways
- Network ACLs
- Network firewalls
- Network interfaces
- VPC route tables
- Security groups
- Target groups
- Transit gateway route tables
- Transit gateway attachments
- VPC interface endpoints
- VPC gateway endpoints
- VPC gateway load balancer endpoints
- VPC endpoints services
- VPC peering connections
- Virtual private gateways

Unsupported resources

- Network Access Analyzer does not produce network paths through resources that are associated with Amazon API Gateway, AWS Global Accelerator, Traffic Mirroring, AWS Wavelength, or AWS Direct Connect. Network Access Analyzer can't produce network paths containing customer managed Amazon EC2 instances that modify packet forwarding behavior.
- Network Access Analyzer doesn't support nested Resource Groups.

Unsupported network configurations

The following network configurations are not supported by Network Access Analyzer.

Internet gateways and virtual private gateways

- Network Access Analyzer supports internet gateways and virtual private gateways at the beginning or end of a path, but does not report paths that pass through internet gateways or virtual private gateways. For example, Network Access Analyzer does not produce paths that start in one VPC, pass through the internet, and end in a second Amazon VPC after passing through an internet gateway. These resources are outside of AWS networking and therefore out of scope.
- Network Access Analyzer does not support NAT reflection at internet gateways. For example, Network Access Analyzer does not report network paths from one network interface to another network interface that are addressed to the second network interface's public IPV4 address.

Application Load Balancers

- Network Access Analyzer does not support [advanced forwarding rules](#).

Gateway Load Balancers

- Packet transformations applied by Gateway Load Balancer targets are ignored. A packet is reflected from the targets back to the Gateway Load Balancer untouched.
- Findings through a Gateway Load Balancer must start at a Gateway Load Balancer endpoint service.

Gateway Load Balancer endpoints

- Paths through a Gateway Load Balancer endpoint do not include the load balancer and its targets.

Network interfaces

- Network Access Analyzer does not produce findings that start or terminate at network interfaces that belong to a NAT gateway or Network Load Balancer.

Network Load Balancers

- Network Access Analyzer does not support instance targets without [client IP preservation](#) enabled.

Network firewalls

- Network Access Analyzer does not analyze network firewall rules. Paths as reported in findings containing network firewalls may be spurious if the firewall on the path is configured with rules that would otherwise block the reported network traffic.

Transit gateways

- Network Access Analyzer does not support paths through AWS Transit Gateway peering connections to other regions or accounts, or AWS Transit Gateway direct connections.

Limitations

- The analysis that Network Access Analyzer performs is limited to IPv4, using UDP or TCP.
- Network Access Analyzer does not report paths that contain resources in accounts or Regions other than the account or Region being analyzed. In particular, Network Access Analyzer does not produce paths containing resources in subnets shared from other accounts, or to resources connected by VPC peering connections, virtual private gateways, internet gateways, or transit gateways to resources in other accounts or in different Regions.
- The paths that Network Access Analyzer reports contain a bounded number of resources. Network Access Analyzer does not produce arbitrary length network paths through atypical

network configurations, such as load balancers that target themselves, or NAT gateways that send packets immediately to another NAT gateway.

- Network Access Analyzer does not ensure that the same findings are produced if you analyze the same Network Access Scope in the same network. Network Access Analyzer might produce new findings for existing Network Access Scope analyses if new configurations are supported in the future.
- Network Access Analyzer reports only unidirectional analysis. That is, Network Access Analyzer findings only indicate that a packet can be sent successfully from a source to a destination.
- Network Access Analyzer does not report connectivity due to traffic mirroring.
- Network Access Analyzer does not consider the health of registered targets.
- Network Access Analyzer does not consider the advertised state of BYOIP address ranges. If a BYOIP address range is not advertised, resources that use these addresses might not be reachable from the internet.
- A running analysis times out after 4 hours.
- Your account has quotas related to Network Access Analyzer. For more information, see [Quotas](#).

Getting started with Network Access Analyzer

You can use Network Access Analyzer to understand network access to resources in your virtual private clouds (VPCs). You can get started with Network Access Analyzer using one of the Amazon created Network Access Scopes.

Tasks

- [Step 1: Analyze your network](#)
- [Step 2: Review your findings](#)
- [Step 3: Delete a Network Access Scope \(Optional\)](#)

Note

Network Access Analyzer evaluates network paths only within the account and Region from which you run the analysis.

Step 1: Analyze your network

To get started quickly, use one of the Network Access Scopes provided by Amazon or create a Network Access Scope using a built-in template. Note that it can take a few minutes to complete the analysis.

To analyze a Network Access Scope

1. Open the Network Manager console at <https://eusc-de-east-1.console.amazonaws-eusc.eu/networkmanager/home>.
2. In the navigation pane, choose **Network Access Analyzer**.
3. If you are using Network Access Analyzer for the first time, choose **Get Started**.
4. Select one of the Amazon created Network Access Scopes:
 - **All-IGW-Ingress (Amazon created)** – Identifies inbound paths from internet gateways to network interfaces.
 - **AWS-IGW-Egress (Amazon created)** – Identifies outbound paths from network interfaces to internet gateways.

- **AWS-VPC-Ingress (Amazon created)** – Identifies inbound paths from internet gateways, peering connections, VPC endpoints, VPNs, and transit gateways to VPCs.
- **AWS-VPC-Egress (Amazon created)** – Identifies outbound paths from VPCs to internet gateways, peering connections, VPC endpoints, VPNs, and transit gateways.

5. Choose **Analyze**.

6. Wait for the analysis to complete and then go to [the section called “Review your findings”](#).

Alternatively, you can get started by creating a Network Access Scope using a built-in template or an empty template.

To create a Network Access Scope

1. Open the Network Manager console at <https://eusc-de-east-1.console.amazonaws-eusc.eu/networkmanager/home>.
2. In the navigation pane, choose **Network Access Analyzer**.
3. Choose **Create Network Access Scope**.
4. Select a built-in template and then choose **Next**.
5. (Optional) Add a [match condition](#).
6. (Optional) Add an [exclusion condition](#).
7. (Optional) To add a tag, choose **Add new tag** and then enter the tag key and tag value.
8. Choose **Next** and then choose **Create Network Access Scope**.
9. Select your Network Access Scope and choose **Analyze**. Wait for the analysis to complete and then go to [the section called “Review your findings”](#).

Step 2: Review your findings

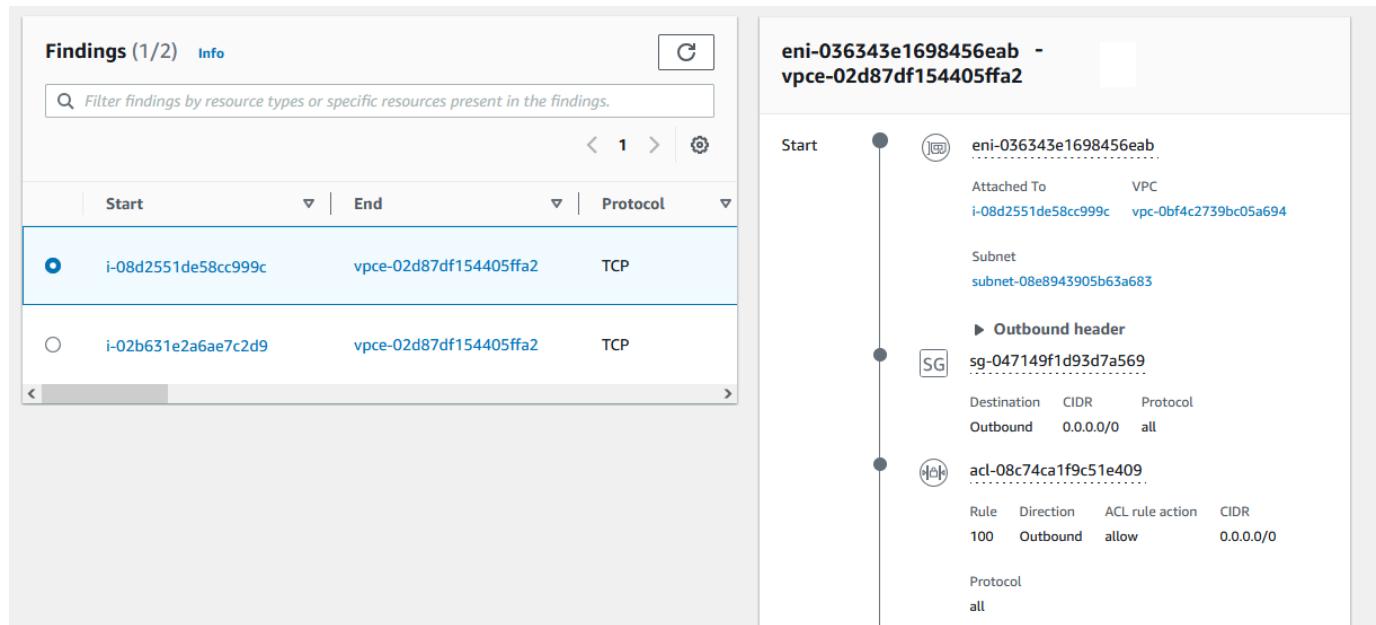
After your analysis is complete, you can review the results.

To review your findings

1. Choose the **Latest analysis** tab. If the analysis produces any findings, **Last analysis result is Findings detected**, as shown in the following figure. Otherwise, **Last analysis result is No findings detected**.

Analysis details				
Analysis ID		Last analysis date	Last analysis result	Analysis status
 nisa-052a0725723061271		November 5, 2024, 8:38 (UTC-08:00)	 Findings detected	 Complete
				Network Interfaces analyzed 21

2. If there are findings detected, the **Findings** pane has the potential network paths identified by the Network Access Scope. You can add filters based on the resources present in the findings. For example, you can filter by resource type.
3. Select a finding to view its details. This information helps you understand the network configurations that produced the finding. For example, you can see the network ACL that applies to traffic that is destined for the internet.



The screenshot shows the Network Access Analyzer interface. On the left, the 'Findings (1/2)' pane displays two findings. The first finding is selected, showing traffic from instance **i-08d2551de58cc999c** to VPC endpoint **vpce-02d87df154405ffa2** via TCP. The second finding shows traffic from instance **i-02b631e2a6ae7c2d9** to the same endpoint. On the right, a detailed view of the selected finding is shown. It shows the network path starting from the instance (eni-036343e1698456eab) and passing through a VPC (i-08d2551de58cc999c, vpc-0bf4c2739bc05a694) and a subnet (subnet-08e8943905b63a683). The path then goes through an SG (sg-047149f1d93d7a569) and an ACL (acl-08c74ca1f9c51e409). The ACL rule 100 is an Outbound rule allowing all traffic to 0.0.0.0/0.

Step 3: Delete a Network Access Scope (Optional)

If you no longer need a Network Access Scope, you can delete it. This action can't be undone.

To delete a Network Access Scope

1. On the Network Access Scopes page, select the check box next to the Network Access Scope.
2. Choose the **Actions** button and then choose **Delete Network Access Scope**.
3. When prompted for confirmation, enter **Delete**.
4. Choose **Delete**.

Getting started with Network Access Analyzer using the AWS CLI

The following procedure describes how to get started with Network Access Analyzer using the AWS CLI.

Tasks

- [Step 1: Create a Network Access Scope](#)
- [Step 2: Analyze a Network Access Scope](#)
- [Step 3: Get the results of a Network Access Scope analysis](#)

Step 1: Create a Network Access Scope

Use the following [create-network-insights-access-scope](#) command to create a Network Access Scope.

```
aws ec2 create-network-insights-access-scope
# optional/example input
--match-paths "Source={ResourceStatement={Resources=vpc-abcd12e3}}"
"Destination={ResourceStatement={ResourceTypes=[\"AWS::EC2::InternetGateway\"]}}"
# optional/example input
--exclude-paths
"Source={ResourceStatement={ResourceTypes=[\"AWS::EC2::InternetGateway\"]}}"
```

The following is example output.

```
{
  "NetworkInsightsAccessScope": {
    "NetworkInsightsAccessScopeId": "nis-0b1889d01c2801311",
    "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-east-1:470889052923:network-
    insights-access-scope/nis-0b1889d01c2801311",
    "CreatedDate": "2024-10-01T13:35:01.017000+00:00",
    "UpdatedDate": "2024-10-01T13:35:01.017000+00:00"
  }
}
```

```
},  
  
"NetworkInsightsAccessScopeContent": {  
  
    "NetworkInsightsAccessScopeId": "nis-0b1889d01c2801311",  
  
    "MatchPaths": [  
  
        {  
  
            "Source": {  
  
                "ResourceStatement": {  
  
                    "Resources": [  
  
                        "vpc-abcd12e3"  
  
                    ]  
  
                }  
  
            }  
  
        },  
  
        {  
  
            "Destination": {  
  
                "ResourceStatement": {  
  
                    "ResourceTypes": [  
  
                        "AWS::EC2::InternetGateway"  
  
                    ]  
  
                }  
  
            }  
  
        }  
  
    }  
  
}
```

```
],  
  
  "ExcludePaths": [  
  
    {  
  
      "Source": {  
  
        "ResourceStatement": {  
  
          "ResourceTypes": [  
  
            "AWS::EC2::InternetGateway"  
  
          ]  
  
        }  
  
      }  
  
    }  
  
  ]  
  
}  
  
}
```

You can also create a scope using the CLI JSON input option, as shown in the following example.

```
aws ec2 create-network-insights-access-scope --cli-input-json file://path-to-access-  
scope-file.json
```

The following is an example input file.

```
{  
  "MatchPaths": [  
    {  
      "Source": {  
        "ResourceStatement": {  
          "Resources": [  
            "vpc-abcd12e3"  
          ]  
        }  
      }  
    }  
  ]  
}
```

```
        }
    }
],
"ExcludePaths": [
{
    "Source": {
        "ResourceStatement": {
            "ResourceTypes": [
                "AWS::EC2::InternetGateway"
            ]
        }
    }
}
]
```

See [Generating an AWS CLI skeleton and input file](#) for more details about using the CLI with JSON input.

Use the following [describe-network-insights-access-scopes](#) command to describe a Network Access Scope.

```
aws ec2 describe-network-insights-access-scopes
```

Use the following [get-network-insights-access-scope-content](#) command to get a Network Access Scope.

```
aws ec2 get-network-insights-access-scope-content --network-insights-access-scope-id
nis-0e123eecc45c67d8
```

Use the following [delete-network-insights-access-scope](#) command to delete a Network Access Scope.

```
aws ec2 delete-network-insights-access-scope --network-insights-access-scope-id
nis-0e123eecc45c67d8
```

Step 2: Analyze a Network Access Scope

Use the following [start-network-insights-access-scope-analysis](#) command to analyze a Network Access Scope. The analysis can take a few minutes to complete.

```
aws ec2 start-network-insights-access-scope-analysis --network-insights-access-scope-id nis-0e123eecc45c67d8
```

The following is example output.

```
{  
  "NetworkInsightsAccessScopeAnalysis": {  
    "NetworkInsightsAccessScopeAnalysisId": "nisa-0e123eecc45c67d89",  
    "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-  
east-1:123456789012:network-insights-access-scope-analysis/nisa-0e123eecc45c67d89",  
    "NetworkInsightsAccessScopeId": "nis-0e123eecc45c67d8",  
    "Status": "running",  
    "StartDate": "2021-11-08T19:29:30.179000+00:00"  
  }  
}
```

Step 3: Get the results of a Network Access Scope analysis

After the analysis completes, you can view the results using the [describe-network-insights-access-scope-analyses](#) command.

```
aws ec2 describe-network-insights-access-scope-analyses
```

Example 1: Success

The following is example output for a successful analysis.

```
        "FindingsFound": "true",
        "Tags": []
    }
]
}
```

Example 2: No findings

The following is example output when no network paths are found in the analysis.

```
aws ec2 get-network-insights-access-scope-analysis-findings --network-insights-access-
scope-analysis-id nisa-07bcaad8bd8160e63
{
    "NetworkInsightsAccessScopeAnalysisId": "nisa-09aeb24f525f2d9f7",
    "AnalysisFindings": []
}
```

Example 3: Findings reported

The following is example output where findings were reported in the analysis.

```
aws ec2 describe-network-insights-access-scope-analyses --network-insights-access-
scope-analysis-id nisa-0c0d3ec68a9bb2f22
{
    "NetworkInsightsAccessScopeAnalyses": [
        {
            "NetworkInsightsAccessScopeAnalysisId": "nisa-09aeb24f525f2d9f7",
            "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope-analysis/nisa-0c0d3ec68a9bb2f22",
            "NetworkInsightsAccessScopeId": "nis-096f763940bb6bcf2",
            "Status": "succeeded",
            "StartDate": "2021-10-06T20:23:53.604000+00:00",
            "FindingsFound": "true",
            "Tags": []
        }
    ]
}
```

```
aws ec2 get-network-insights-access-scope-analysis-findings --network-insights-access-
scope-analysis-id nisa-0c0d3ec68a9bb2f22 --max-results 1
{
    "NetworkInsightsAccessScopeAnalysisId": "nisa-09aeb24f525f2d9f7",
    "AnalysisFindings": [

```

```
{  
  "NetworkInsightsAccessScopeAnalysisId": "nisa-09aeb24f525f2d9f7",  
  "NetworkInsightsAccessScopeId": "nis-096f763940bb6bcf2",  
  "FindingComponents": [  
    {  
      "SequenceNumber": 1,  
      "Component": {  
        "Id": "igw-1a23b4cd",  
        "Arn": "arn:aws:ec2:us-east-1:123456789012:internet-gateway/  
igw-1a23b4cd"  
      },  
      "OutboundHeader": {  
        "DestinationAddresses": [  
          "172.31.22.225/32"  
        ]  
      },  
      "InboundHeader": {  
        "DestinationAddresses": [  
          "52.2.112.57/32"  
        ],  
        "DestinationPortRanges": [  
          {  
            "From": 80,  
            "To": 80  
          }  
        ],  
        "Protocol": "6",  
        "SourceAddresses": [  
          "0.0.0.0/5",  
          "11.0.0.0/8",  
          "12.0.0.0/6",  
          "128.0.0.0/3",  
          "16.0.0.0/4",  
          "160.0.0.0/5",  
          "168.0.0.0/6",  
          "172.0.0.0/12",  
          "172.128.0.0/9",  
          "172.32.0.0/11",  
          "172.64.0.0/10",  
          "173.0.0.0/8",  
          "174.0.0.0/7",  
          "176.0.0.0/4",  
          "192.0.0.0/9",  
          "192.128.0.0/11",  
        ]  
      }  
    }  
  ]  
}
```

```
        "192.160.0.0/13",
        "192.169.0.0/16",
        "192.170.0.0/15",
        "192.172.0.0/14",
        "192.176.0.0/12",
        "192.192.0.0/10",
        "193.0.0.0/8",
        "194.0.0.0/7",
        "196.0.0.0/6",
        "200.0.0.0/5",
        "208.0.0.0/4",
        "224.0.0.0/3",
        "32.0.0.0/3",
        "64.0.0.0/2",
        "8.0.0.0/7"
    ],
    "SourcePortRanges": [
        {
            "From": 0,
            "To": 65535
        }
    ]
},
{
    "SequenceNumber": 2,
    "AclRule": {
        "Cidr": "0.0.0.0/0",
        "Egress": false,
        "Protocol": "all",
        "RuleAction": "allow",
        "RuleNumber": 100
    },
    "Component": {
        "Id": "acl-579af131",
        "Arn": "arn:aws:ec2:us-east-1:123456789012:network-acl/acl-579af131"
    }
},
{
    "SequenceNumber": 3,
    "Component": {
        "Id": "sg-0cab31773e042794f",

```

```

        "Arn": "arn:aws:ec2:us-east-1:123456789012:security-group/
sg-0cab31773e042794f"
    },
    "SecurityGroupRule": {
        "Cidr": "0.0.0.0/0",
        "Direction": "ingress",
        "PortRange": {
            "From": 80,
            "To": 80
        },
        "Protocol": "tcp"
    }
},
{
    "SequenceNumber": 4,
    "Component": {
        "Id": "eni-0680af09e502660e7",
        "Arn": "arn:aws:ec2:us-east-1:123456789012:network-interface/
eni-0680af09e502660e7"
    },
    "Subnet": {
        "Id": "subnet-8061f9db",
        "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-8061f9db"
    },
    "Vpc": {
        "Id": "vpc-abcd12e3",
        "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-abcd12e3"
    }
}
],
},
"NextToken": "AYADeDdyvQENR4bFEGARVcz0dwQAhwACABFFbmNyeXB0aW9uQ29udGV4dAATVG9rZW5FbmNyeXB0aW9uVXRpbAAVYXdzL
+v6C/JyLKMZzcGXs3NAp676D8RwoAdF/
sSFYUnAA7JwYLP1YSfBZ5fHHPjJ8Y6AVkJEzpGGza1CuzHFG9dqvkyuLoYxkpqGgbv0e0T2Q0rLfJID
+vNWEqSb03/6JX1tR5ipYGD7yAn0b6vCBmheU9dDdbPE1SnidTc6XLpR8ihzdqSaJZns1AxYXNcsjrSEWmERdBh0IBaUUhR
+1zU9BgN/NrgBnMGUCMQDSA4E1zrjcR
+iFS4RNJincDtRKZz3T2AmoI23+Xh440HSrTR2XgBdewZZzvKX1tdkCMHDGRfeLrJMXLvVo/
sHL6ZqGR1FYWs3UWhMpkMGDdXZcQL+is60dXqAY1L0JLaDpaQ=="
}

```

Note

The list of source addresses in the previous example includes everything in the 0.0.0.0/0 address range except for the RFC1918 range.

Network Access Scopes in Network Access Analyzer

With Network Access Analyzer, you can specify your network access requirements by using Network Access Scopes. A Network Access Scope defines outbound and inbound traffic patterns, including sources, destinations, paths, and traffic types. Each Network Access Scope consists of one or more match conditions, and zero or more exclusion conditions.

When you start an analysis on a Network Access Scope, Network Access Analyzer produces findings. It identifies network paths in the Network Access Scope that match at least one of the match conditions, and none of the exclude conditions. By combining match and exclude conditions, you can refine the findings produced by Network Access Analyzer to identify unexpected connectivity in your network.

Match and exclude conditions have similar structures. They consist of resource statements and packet header statements that specify the network traffic to match or exclude.

Contents

- [Network Access Analyzer resource statements](#)
- [Packet header statements in Network Access Analyzer](#)
- [Match conditions in Network Access Analyzer](#)
- [Exclusion conditions in Network Access Analyzer](#)
- [Example Network Access Scopes in Network Access Analyzer](#)

Network Access Analyzer resource statements

A resource statement in Network Access Analyzer defines the network components for a match or exclude condition. Each resource statement includes resource IDs, resource ARNs, or resource types. A single resource statement can include either resource IDs or resource types, but not both.

You can specify the following components by resource ID or resource ARN:

- EC2 instances (source and destination only)
- Internet gateways (source and destination only)
- NAT gateways (through only)
- Network firewalls (through only)
- Network interfaces (source and destination only)

- Resource groups
- Security groups (source and destination only)
- Subnets (source and destination only)
- Transit gateway attachments
- Virtual private clouds (VPC) (source and destination only)
- Virtual private gateways (source and destination only)
- VPC endpoint services
- VPC endpoints
- VPC peering connections

You must specify the following components by ARN:

- Classic, Application, Network, and Gateway Load Balancers (through only)

You can specify the following components by resource type:

- AWS::EC2::InternetGateway (source and destination only)
- AWS::EC2::NatGateway (through only)
- AWS::EC2::TransitGatewayAttachment
- AWS::EC2::VPCEndpoint (destination and through only)
- AWS::EC2::VPCEndpointService
- AWS::EC2::VPCPeeringConnection
- AWS::EC2::VPNGateway (source and destination only)
- AWS::ElasticLoadBalancing::LoadBalancer (through only)
- AWS::ElasticLoadBalancingV2::LoadBalancer (through only)
- AWS::NetworkFirewall::NetworkFirewall (through only)

Packet header statements in Network Access Analyzer

A packet header statement defines the traffic types for a match or exclude condition. If you omit the packet header statement, all traffic types match. All fields are optional, but if you use a packet header statement, you must use at least one of its fields.

You can specify the following fields:

- **Protocols** – The protocol strings to match. The possible values are `tcp` and `udp`. You can specify one of the values or both of the values. If you omit this field, packets with either the `tcp` or `udp` protocol are admitted.
- **SourceAddress** – The IP addresses or CIDR ranges. You can't specify this option with **SourcePrefixLists**. If specified, only packets with matching source addresses are admitted. If you don't specify **SourcePrefixLists** or **SourceAddresses**, packets with any source address are admitted.
- **SourcePrefixLists** – The IDs or ARNs of the prefix lists. You can't specify this option with **SourceAddresses**. If specified, only packets with matching source addresses are admitted. If you don't specify **SourcePrefixLists** or **SourceAddresses**, packets with any source address are admitted.
- **DestinationAddress** – The IP addresses or CIDR ranges. This option is mutually exclusive with **DestinationPrefixLists**. If specified, only packets with matching destination addresses are admitted. If you don't specify **DestinationPrefixLists** or **DestinationAddress**, packets with any destination address are admitted.
- **DestinationPrefixLists** – The IDs or ARNs of the prefix lists. This option is mutually exclusive with **DestinationAddress**. If specified, only packets with matching destination addresses are admitted. If you don't specify **DestinationPrefixLists** or **DestinationAddress**, packets with any destination address are admitted.
- **SourcePorts** – The ports or port ranges. If specified, only packets with source ports that match one of the ports or port ranges are admitted. If omitted, packets with any source port are admitted.
- **DestinationPorts** – The ports or port ranges. If specified, only packets with destination ports that match one of the ports or ranges are admitted. If omitted, packets with any destination port are admitted.

Match conditions in Network Access Analyzer

A match condition defines the types of network paths that should be produced as findings. A Network Access Scope must specify at least one match condition. A match condition can contain a source and a destination. Each source and destination can include a resource statement, a packet header statement, or both.

If a match condition has a source but no destination, it produces findings for the following:

- Network paths that end at any supported resource
- Network paths that start at a network component specified in the resource statement of the source (if defined)
- Network paths with a packet header that matches the packet header statement of the source (if defined)

If a match condition has a destination but no source, it produces findings for the following:

- Network paths that start at any supported resource and end at a network component specified in the resource statement of the destination (if defined)
- Network paths with a packet header that matches the packet header statement of the destination (if defined)

If a match condition has both a source and destination, the network path must start at the source entry and end at the destination.

If a Network Access Scope has multiple match conditions, it produces findings for any path that satisfies at least one of the match conditions.

Exclusion conditions in Network Access Analyzer

A Network Access Scope produces findings only for paths that match at least one match condition, but do not match any exclusion conditions.

An exclusion condition can contain source, destination, and through fields. Each field is optional, but you must specify at least one field. Each source and destination can include a resource statement, a packet header statement, or both.

A through entry contains exactly one element that contains a resource statement. It excludes paths that contain the specified network component anywhere along the path, not just at the beginning or end. You can use a through entry in combination with a source, a destination, or both a source and a destination.

Example Network Access Scopes in Network Access Analyzer

The following are examples of Network Access Scopes.

Example: Identify all traffic between two subnets

To identify traffic between two subnets that are intended to be isolated from each other, create an access scope with two match conditions. The first condition identifies paths from network interfaces in the first subnet to network interfaces in the second subnet. The second condition identifies paths from network interfaces in second subnet to network interfaces in the first subnet.

```
{  
  "MatchPaths": [  
    {  
      "Source": {  
        "ResourceStatement": {  
          "Resources": [ "subnet-1-id" ]  
        }  
      },  
      "Destination": {  
        "ResourceStatement": {  
          "Resources": [ "subnet-2-id" ]  
        }  
      }  
    },  
    {  
      "Source": {  
        "ResourceStatement": {  
          "Resources": [ "subnet-2-id" ]  
        }  
      },  
      "Destination": {  
        "ResourceStatement": {  
          "Resources": [ "subnet-1-id" ]  
        }  
      }  
    }  
  ]  
}
```

Example: Use resource groups with Network Access Scopes

To identify paths to or from resources with specific resource tags, you can use AWS resource groups. With resource groups, you define a set of resources that contain a specific tag. For more information, see [Build a tag-based query and create a group](#).

The following example identifies inbound paths to any Amazon EC2 instances in the specified resource group.

```
{  
  "MatchPaths": [  
    {  
      "Destination": {  
        "ResourceStatement": {  
          "Resources": [  
            "arn:aws:resource-groups:us-east-1:123456789012:group/bastions"  
          ]  
        }  
      }  
    }  
  ]  
}
```

To identify inbound paths to bastion hosts that use a port other than port 22 (SSH), combine the match condition in the previous example with an exclude condition that specifies destination port 22.

```
{  
  "MatchPaths": [  
    {  
      "Destination": {  
        "ResourceStatement": {  
          "Resources": [  
            "arn:aws:resource-groups:us-east-1:123456789012:group/bastions"  
          ]  
        }  
      }  
    }  
  ],  
  "ExcludePaths": [  
    {  
      "Destination": {  
        "PacketHeaderStatement": {  
          "DestinationPorts": [  
            "22"  
          ]  
        }  
      }  
    }  
  ]  
}
```

```
    ]  
}
```

Example: Identify all inbound traffic from the public internet, except for a trusted CIDR range

The following example identifies traffic from the internet while excluding paths that start from a trusted address range.

```
{  
    "MatchPaths": [  
        {  
            "Source": {  
                "ResourceStatement": {  
                    "ResourceTypes": [  
                        "AWS::EC2::InternetGateway"  
                    ]  
                }  
            }  
        }  
    ],  
    "ExcludePaths": [  
        {  
            "Source": {  
                "PacketHeaderStatement": {  
                    "SourceAddresses": [  
                        "55.3.0.0/16"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

Example: Exclude traffic that originates at the addresses in a prefix list

The following example identifies traffic to the public internet while excluding traffic that originates at the addresses in the specified prefix list.

```
{  
    "MatchPaths": [  
        {  
            "Source": {
```

```
        "ResourceStatement": {
            "ResourceTypes": [
                "AWS::EC2::InternetGateway"
            ]
        }
    }
],
"ExcludePaths": [
{
    "Source": {
        "PacketHeaderStatement": {
            "SourcePrefixLists": [
                "pl-02cd2c6b"
            ]
        }
    }
}
]
```

Example: Identify all outbound traffic to the public internet, excluding a trusted CIDR range

The following example identifies traffic to the public internet while excluding a trusted address range.

```
{
    "MatchPaths": [
        {
            "Destination": {
                "ResourceStatement": {
                    "ResourceTypes": [
                        "AWS::EC2::InternetGateway"
                    ]
                }
            }
        }
    ],
    "ExcludePaths": [
        {
            "Destination": {
                "PacketHeaderStatement": {
                    "DestinationAddresses": [

```

```
        "55.3.0.0/16"
    ]
}
}
]
}
```

Example: Identify inbound traffic that bypasses a network firewall

The following example identifies inbound traffic to network interfaces in a subnet that bypasses a network firewall.

```
{
  "MatchPaths": [
    {
      "Source": {
        "ResourceStatement": {
          "ResourceTypes": [
            "AWS::EC2::InternetGateway"
          ]
        }
      },
      "Destination": {
        "ResourceStatement": {
          "Resources": [
            "subnet-814424dd"
          ]
        }
      }
    }
  ],
  "ExcludePaths": [
    {
      "ThroughResources": [
        {
          "ResourceStatement": {
            "ResourceTypes": [
              "AWS::NetworkFirewall::Firewall"
            ]
          }
        }
      ]
    }
  ]
}
```

```
    }
]
}
```

Example: Identify traffic to network interface with a specific security group

The following example identifies traffic to network interfaces with a specific security group.

```
{
  "MatchPaths": [
    {
      "Source": {
        "ResourceStatement": {
          "ResourceTypes": [
            "AWS::EC2::InternetGateway"
          ]
        }
      },
      "Destination": {
        "ResourceStatement": {
          "Resources": [
            "sg-f15d59b3"
          ]
        }
      }
    }
  ]
}
```

Identity and access management for Network Access Analyzer

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 Network Access Analyzer resources. IAM is an AWS service that you can use with no additional charge.

Contents

- [Audience](#)
- [Authenticating with identities](#)
- [Managing access using policies](#)
- [How Network Access Analyzer works with IAM](#)
- [Required API permissions for Network Access Analyzer](#)
- [AWS managed policies for Network Access Analyzer](#)

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Network Access Analyzer.

Service user – If you use the Network Access Analyzer service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Network Access Analyzer features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator.

Service administrator – If you're in charge of Network Access Analyzer resources at your company, you probably have full access to Network Access Analyzer. It's your job to determine which Network Access Analyzer features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Network Access Analyzer.

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 Network Access Analyzer works with IAM

Before you use IAM to manage access to Network Access Analyzer, learn what IAM features are available to use with Network Access Analyzer.

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

To get a high-level view of how Network Access Analyzer 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 Network Access Analyzer

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 Network Access Analyzer

Supports resource-based policies: No

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

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

Policy actions for Network Access Analyzer

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.

Network Access Analyzer shares its API namespace with Amazon EC2. Policy actions in Network Access Analyzer use the following prefix before the action:

```
ec2
```

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

```
"Action": [
    "ec2:action1",
    "ec2:action2"
]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action.

```
"Action": "ec2:Describe*"
```

The following actions are supported by Network Access Analyzer:

- `CreateNetworkInsightsAccessScope`
- `DeleteNetworkInsightsAccessScope`
- `DeleteNetworkInsightsAccessScopeAnalysis`
- `DescribeNetworkInsightsAccessScopeAnalyses`
- `DescribeNetworkInsightsAccessScopes`
- `GetNetworkInsightsAccessScopeAnalysisFindings`
- `GetNetworkInsightsAccessScopeContent`
- `StartNetworkInsightsAccessScopeAnalysis`

For more information, see [Actions Defined by Amazon EC2](#) in the *Service Authorization Reference*.

Policy resources for Network Access Analyzer

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": "*"
```

The following Network Access Analyzer API actions do not support resource-level permissions:

- `DescribeNetworkInsightsAccessScopeAnalyses`
- `DescribeNetworkInsightsAccessScopes`

Policy condition keys for Network Access Analyzer

Supports service-specific policy condition keys: No

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*.

ACLs in Network Access Analyzer

Supports ACLs: No

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

ABAC with Network Access Analyzer

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*.

Using temporary credentials with Network Access Analyzer

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 Network Access Analyzer

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 Network Access Analyzer

Supports service roles: No

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*.

Service-linked roles for Network Access Analyzer

Supports service-linked roles: No

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.

Required API permissions for Network Access Analyzer

Network Access Analyzer relies on data from other AWS services. It uses permissions from the following services:

- Amazon EC2

- ELB
- AWS Network Firewall
- AWS Resource Groups
- AWS Resource Groups Tagging API
- AWS Tiros

To view the permissions for this policy, see [AmazonVPCNetworkAccessAnalyzerFullAccessPolicy](#) in the *AWS Managed Policy Reference*.

Additional information

Network Access Analyzer API calls

The following permissions are required to call the Network Access Analyzer APIs. Users need these permissions to create and start analyzing Network Access Scopes, or to view and delete existing paths and analyses in your account. You must grant users permission to call the Network Access Analyzer API actions that they need.

- `ec2:CreateNetworkInsightsAccessScope`
- `ec2:DeleteNetworkInsightsAccessScope`
- `ec2:DeleteNetworkInsightsAccessScopeAnalysis`
- `ec2:DescribeNetworkInsightsAccessScopeAnalyses`
- `ec2:DescribeNetworkInsightsAccessScopes`
- `ec2:GetNetworkInsightsAccessScopeAnalysisFindings`
- `ec2:GetNetworkInsightsAccessScopeContent`
- `ec2:StartNetworkInsightsAccessScopeAnalysis`

Describe API calls for networking-related resources

Network Access Analyzer uses describe calls while gathering information about your resources from Amazon VPC, Amazon EC2, ELB, and AWS Network Firewall (for example, subnets, network interfaces, and security groups). To access Network Access Analyzer, users must also have these API permissions.

If you specify a resource group in a resource statement, Network Access Analyzer uses `resource-groups>ListResourceGroups` while gathering information about your network configuration.

This action requires the following permissions: `cloudformation:DescribeStacks` `cloudformation>ListStackResources`, and `tag:GetResources`.

Tagging-related API calls

To tag or untag Network Access Analyzer resources, users need the following Amazon EC2 API permissions. To allow users to work with tags, you must grant them permission to use the specific tagging actions that they need.

- `ec2:CreateTags`
- `ec2:DeleteTags`

Tiros API calls

If you monitor API calls, you might see calls to Tiros APIs. Tiros is a service that is only accessible by AWS services and that surfaces network findings to Network Access Analyzer. Calls to the Tiros endpoint are required for Network Access Analyzer to function. To access Network Access Analyzer, users must also have the same API permissions.

AWS managed policies for Network Access Analyzer

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to [create IAM customer managed policies](#) that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see [AWS managed policies](#) in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new

operations and resources. For a list and descriptions of job function policies, see [AWS managed policies for job functions](#) in the *IAM User Guide*.

AWS managed policy:

AmazonVPCNetworkAccessAnalyzerFullAccessPolicy

Provides permissions to create, analyze, and delete Network Access Scopes, and to describe network path resources, such as firewalls, internet gateways, load balancers, NAT gateways, network interfaces, transit gateway attachments, VPC endpoints, VPC peering connections, and virtual private gateways.

To view the permissions for this policy, see [AmazonVPCNetworkAccessAnalyzerFullAccessPolicy](#) in the *AWS Managed Policy Reference*.

Network Access Analyzer does not support resources from Direct Connect (service prefix: `directconnect`) or AWS Global Accelerator (service prefix: `globalaccelerator`). If you use this policy as a model for your own policies, you can omit these actions.

Network Access Analyzer updates to AWS managed policies

View details about updates to AWS managed policies for Network Access Analyzer since this service began tracking these changes.

Change	Description	Date
AmazonVPCNetworkAccessAnalyzerFullAccessPolicy – Update to an existing policy	Added the action <code>elasticloadbalancing:DescribeTargetGroupAttributes</code> , which grants permission to describe the attributes of a target group.	May 15, 2024
AmazonVPCNetworkAccessAnalyzerFullAccessPolicy – Update to an existing policy	Removed resource ID prefixes from the resource ARNs used to allow tagging Network Access Analyzer resources on create.	November 3, 2023
AmazonVPCNetworkAccessAnalyzerFullAccessPolicy – New policy	Added a policy that provides full access to Network Access Analyzer.	June 15, 2023

Change	Description	Date
Network Access Analyzer started tracking changes	Network Access Analyzer started tracking changes for its AWS managed policies.	December 1, 2021

Quotas and considerations for Network Access Analyzer

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. You can request increases for some quotas, but not for all quotas.

To view the quotas for Network Access Analyzer, open the [Service Quotas console](#). In the navigation pane, choose **AWS services**, and then select **Network Insights**. To request a quota increase, see [Requesting a quota increase](#) in the *Service Quotas User Guide*.

Your AWS account has the following quotas related to Network Access Analyzer.

Name	Default	Adjustable
Access scopes	1,000	Yes
Access scope analyses	10,000	Yes
Concurrent access scope analyses	25	Yes
Findings per scope analysis	10,000	No

Analysis runtime

All network interfaces in the account and Region are included in every analysis. The running analysis times out after 4 hours.

Troubleshooting Network Access Analyzer

The following error messages are returned by Network Access Analyzer:

The request failed due to insufficient permissions

Verify that you have the required permissions. For more information, see [the section called "Required API permissions"](#).

The network configuration is not supported

Verify that you are using resources that are supported by Network Access Analyzer. For more information, see [the section called "Supported path resources"](#).

The request failed due to modifications in network resources during the analysis.

You can't update your network while the analysis is running.

The request failed due to missing component [*component*]

Verify that the resource ARNs are correct. For more information, see the [Service Authorization Reference](#).

The request failed due to inaccessible resource [*resource*]

Verify that you have permission to access the specified resource.

The request failed due to throttling errors from [*service*]

Check for other applications or services that are currently consuming read capacity for the specified service.

Document history for Network Access Analyzer

The following table describes the releases for Network Access Analyzer.

Change	Description	Date
<u>AWS managed policy updates</u>	Network Access Analyzer updated one existing policy.	May 15, 2024
<u>AWS managed policy updates</u>	Network Access Analyzer updated one existing policy.	November 3, 2023
<u>AWS managed policy updates</u>	Network Access Analyzer added one new policy.	June 15, 2023
<u>Initial release</u>	This release introduces Network Access Analyzer.	December 1, 2021