# Amazon Virtual Private Cloud

## Reachability Analyzer

# Amazon Virtual Private Cloud: Reachability Analyzer

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

# Table of Contents

# What is Reachability Analyzer?

Reachability Analyzer is a configuration analysis tool that enables you to perform connectivity testing between a source resource and a destination resource in your virtual private clouds (VPCs). When the destination is reachable, Reachability Analyzer produces hop-by-hop details of the virtual network path between the source and the destination. When the destination is not reachable, Reachability Analyzer identifies the blocking component. For example, paths can be blocked by configuration issues in a security group, network ACL, route table, or load balancer.

For more information, see How Reachability Analyzer works (p. 2).

## Use cases

You can use Reachability Analyzer to do the following:

- Troubleshoot connectivity issues caused by network misconfiguration.
- Verify that your network configuration matches your intended connectivity.
- Automate the verification of your connectivity intent as your network configuration changes.

## Working with Reachability Analyzer

You can use any of the following interfaces to work with Reachability Analyzer:

- **AWS Management Console** — A web interface for AWS services, including Reachability Analyzer.
- **AWS Command Line Interface (AWS CLI)** — Provides commands for AWS services, including Reachability Analyzer. The AWS CLI is supported on Windows, macOS, and Linux. For more information, see the AWS Command Line Interface User Guide.
- **AWS CloudFormation** — Enables you to create templates that describe your AWS resources. You use a template to provision and manage AWS resources as a single unit. For more information, see the following resources: AWS::EC2::NetworkInsightsAnalysis and AWS::EC2::NetworkInsightsPath.
- **AWS SDKs** — Provides language-specific APIs and takes care of many of the connection details, such as calculating signatures, handling request retries, and handling errors. For more information, see AWS SDKs.
- **Query API** — Provides low-level API actions that you call using HTTPS requests. Using the Query API is the most direct way to access Reachability Analyzer. However, the Query API requires that your application handle low-level details such as generating the hash to sign the request, and handling errors. For more information, see the Amazon EC2 API Reference.

## Pricing

You are charged per analysis run between a source and destination. For more information, see Pricing.

# How Reachability Analyzer works

Reachability Analyzer analyzes the path between a source and destination by building a model of the network configuration, and then checking for reachability based on the configuration. It does not send packets or analyze the data plane.

To use Reachability Analyzer, you specify the path for the traffic from a source to a destination. For example, you could specify an internet gateway as the source, an EC2 instance as the destination, 22 as the destination port, and TCP as the protocol. This would allow you to verify that you can connect to the EC2 instance through the internet gateway using SSH.

If there are multiple reachable paths between a source and a destination, Reachability Analyzer identifies and displays the shortest path. You can analyze the path again, specifying an intermediate component, to find an alternative reachable path that traverses the intermediate component.

If the path is not reachable, Reachability Analyzer displays information about the component or combination of components that is blocking the path. There might be additional components blocking the path.

**Contents**



## Source and destination resources

The source and destination resources must be in the same Region. The source and destination resources must be in the same VPC or in VPCs that are connected through a VPC peering connection or a transit gateway. The source and destination resources can belong to different AWS accounts in the same organization from AWS Organizations.

Reachability Analyzer supports the following resource types as sources and destinations:

- Instances
- Internet gateways
- Network interfaces
- Transit gateways
- Transit gateway attachments
- VPC endpoint services
- VPC endpoints
- VPC peering connections
- VPN gateways

In addition, Reachability Analyzer supports IP addresses as destinations.

# Intermediate components

Reachability Analyzer supports the following resource types as intermediate components:

- Load balancers
- NAT gateways
- Network Firewall firewall
- Transit gateways
- Transit gateway attachments
- VPC peering connections

# Path components

The following resource types can appear in reachable paths and in explanations when a path is not reachable:

- EC2 instances
- Internet gateways
- Load balancers
- NAT gateways
- Network ACLs
- Network Firewall firewall
- Network interfaces
- Prefix lists
- Route tables
- Security groups
- Subnets
- Target groups
- Transit gateways
- Transit gateway attachments
- Transit gateway route tables
- Virtual private gateways
- VPC endpoint services
- VPC endpoints
- VPC gateway endpoints
- VPC peering connections
- VPCs
- VPN connections

# Considerations

Consider the following when working with Reachability Analyzer:

- Reachability Analyzer supports only resources with an IPv4 address. If a resource has both IPv4 and IPv6 addresses, Reachability Analyzer includes only the IPv4 addresses in its analysis.

- Reachability Analyzer supports shared resources only if they can be fully described by the calling principal. For example, if a route references a prefix list owned by another account, the owner must share the prefix list with the calling principal for the analysis to succeed.
- If you enable trusted access, the delegated administrator account can create and delete paths that traverse owner and participant subnets within your organization from AWS Organizations. This account can also start and delete path analyses. For more information, see *Cross-account analyses* (p. 28).
- Paths are not a shareable resource.
- Transit gateway Connect attachments are not supported. Reachability Analyzer analyzes connectivity only up to these attachments.
- With the TCP protocol, when a network path traverses a transit gateway route table, only forward traffic is analyzed.
- Reachability Analyzer can find paths through at most two transit gateway route tables. To analyze paths through additional transit gateway route tables, use Route Analyzer. For more information, see Route Analyzer in the *Amazon VPC Transit Gateways* guide.
- Paths through a Gateway Load Balancer endpoint do not include the Gateway Load Balancer or its targets. You should verify connectivity between the Gateway Load Balancer and its targets using a separate analysis.
- Reachability Analyzer does not support Network Firewall rule groups that reference a resource group. In this case, the analysis fails.
- Reachability Analyzer supports all stateful and stateless 5-tuple rules in Network Firewall. It doesn't support domain lists, Suricata rules, rule options, and tag-based resource groups. When Reachability Analyzer encounters an unsupported rule in Network Firewall, it provides an informational message in the path details.
- The packet header leaving the source and the packet header arriving at the destination can differ, due to intermediate components transforming the packets. For example, internet gateways and NAT gateways provide network address translation (NAT).
- Reachability Analyzer automatically deletes an analysis 120 days after its creation date.

# Resource configuration

Use the following documentation to help you update the configuration of your network resources:

- Elastic network interfaces
- Firewalls (AWS Network Firewall)
- Internet gateways
- Load balancers and target groups (Elastic Load Balancing)
  - Application Load Balancers
  - Classic Load Balancers
  - Gateway Load Balancers
  - Network Load Balancers
- Network ACLs
- Route tables
- Security groups for Linux instances
- Security groups for Windows instances
- Transit gateways
- VPC endpoint services (AWS PrivateLink)
- VPC peering configurations
- VPN connections

# Getting started with Reachability Analyzer

You can use Reachability Analyzer to determine whether a destination resource in your virtual private cloud (VPC) is reachable from a source resource. To get started, you specify a source and a destination. For example, you can run a reachability analysis between two network interfaces or between a network interface and a gateway. If there is a reachable path between the source and destination, Reachability Analyzer displays the details. Otherwise, Reachability Analyzer identifies the blocking component.

**Tasks**



## Step 1: Create and analyze a path

Specify the path for the traffic from a source to a destination. After you create the path, Reachability Analyzer analyzes the path once. You can analyze a path at any time to determine whether your intended connectivity is supported, even as your network configuration changes.

**To create a path**

1. Open the Network Manager console at https://console.aws.amazon.com/networkmanager/home.
2. In the navigation pane, choose **Reachability Analyzer**.
3. Choose **Create and analyze path**.
4. (Optional) For **Name tag**, enter a descriptive name for the analysis.
5. To specify the source resource, choose the resource type from **Source type**, and then choose the specific resource from **Source**.

   (Optional) You can filter the scope of the result based on the packet header leaving the source resource. For example, use the source and destination IP addresses and ports of interest. By default, the analysis considers all combinations of IP addresses and ports.
6. To specify the destination resource, choose the resource type from **Destination type**, and then choose the specific resource from **Destination**.

   (Optional) You can filter the scope of the result based on the packet header arriving at the destination resource. For example, use the source and destination IP addresses and ports of interest. By default, the analysis considers all combinations of IP addresses and ports.
7. For **Protocol**, choose **TCP** or **UDP**.
8. (Optional) To add a tag, choose **Add new tag** and then enter the tag key and tag value.
9. Choose **Create and analyze path**.

## Step 2: View the results of the path analysis

After the path analysis completes, you can view the result of the analysis.

**To view the results of the path analysis**

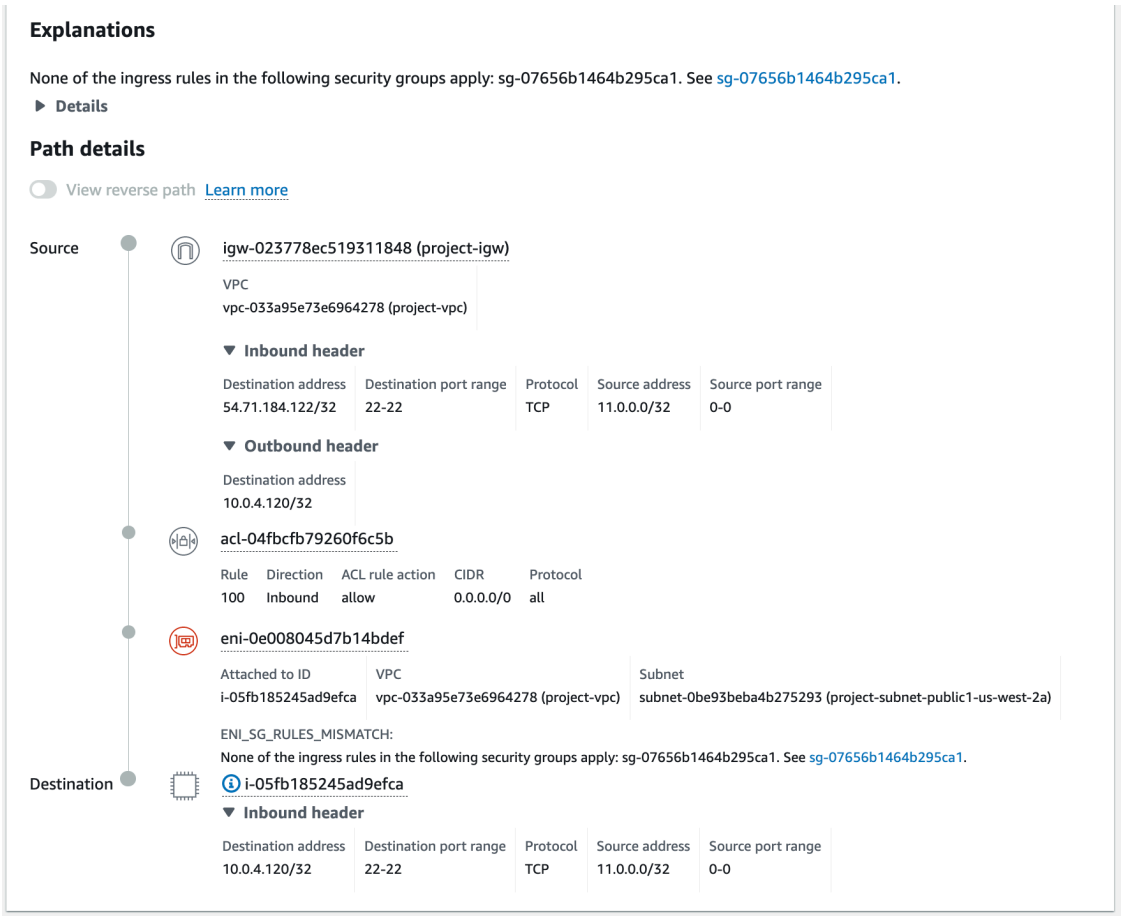
1. Choose the ID of the path in the **Path ID** column to view the path details page.
2. In the **Analysis explorer** panel, find **Reachability status** and check whether it is **Reachable** or **Not reachable**. If the path is reachable, the console displays the shortest route found between the source and destination. Otherwise, expand **Explanations**, **Details** for information about the blocking component.
3. If the reachability status matches your intent, there is no further action required. Consider running the analysis again if you change your network configuration so that you can ensure that the reachability status still matches your intent. Otherwise, proceed to Step 3 (p. 6).

# Step 3: Change the network configuration and analyze the path

If the reachability status does not match your intent, you can change your network configuration. Then you can analyze the path again to confirm that the reachability status matches your intent.

**To restore connectivity for a path that is not reachable**

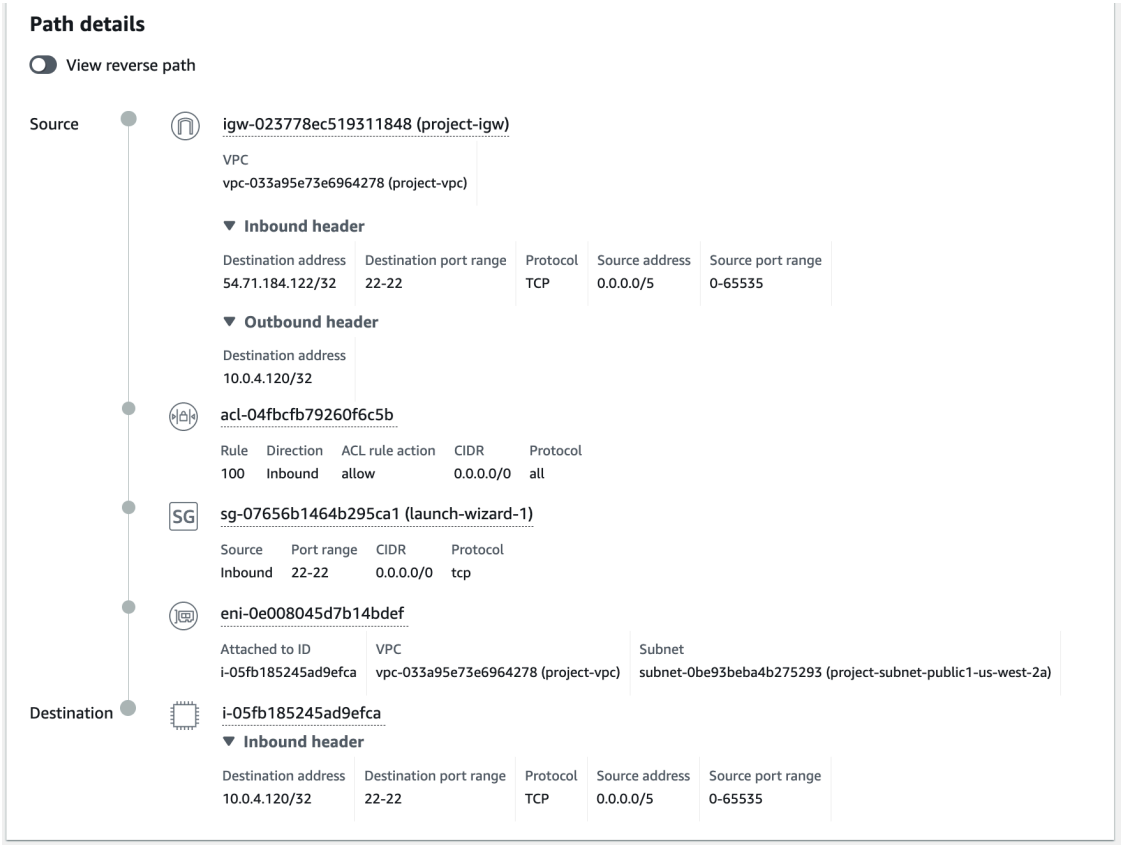
1. The **Analysis explorer** panel includes an explanation code (p. 19) and detailed information about the component or combination of components that is blocking the path (under **Explanations**, **Details**). For example, in the following explanation, a security group is missing a required inbound rule.



2. Update the configuration of the component so that the desired traffic can traverse the component.
3. Choose **Analyze path** to confirm that the path is now reachable. You can optionally specify the Amazon Resource Name (ARN) of a resource that the path must traverse.

### To remove connectivity for a reachable path

1. The **Analysis explorer** panel includes a visual representation of the shortest route found between the source and destination. It includes all components between the source and destination. For example, the following diagram shows the components that traffic traverses from the source internet gateway to the destination EC2 instance.




2. Identify the component that is overly permissive and update its configuration.
3. Choose **Analyze path** to confirm that the path is no longer reachable.

# Step 4: Delete the path

If you no longer need the path, you can delete it. When you delete a path, you also delete all its analyses. If you keep the path, note that Reachability Analyzer will automatically delete the analysis 120 days after its creation date.

### To delete the path

1. Open the Network Manager console at https://console.aws.amazon.com/networkmanager/home.
2. In the navigation pane, choose **Reachability Analyzer**.
3. Select the path.

4. Choose **Actions**, **Delete path**.
5. When prompted for confirmation, choose **Delete path**.

# Getting started with Reachability Analyzer using the AWS CLI

You can use Reachability Analyzer to determine whether a destination resource in your virtual private cloud (VPC) is reachable from a source resource. To get started, you specify a source and a destination. For example, you can run a reachability analysis between two network interfaces or between a network interface and a gateway. If there is a reachable path between the source and destination, Reachability Analyzer displays the details. Otherwise, Reachability Analyzer identifies the blocking component.

**Tasks**



## Step 1: Create a path

Use the following create-network-insights-path command to create a path. In this example, the source is an internet gateway and the destination is an EC2 instance.

```
aws ec2 create-network-insights-path
    --source igw-0797cccdc9d73b0e5
    --destination i-0495d385ad28331c7
    --protocol TCP
    --filter-at-source file://source-filter.json
```

The following is an example `source-filter.json`.

```
{
    "DestinationPortRange": {
        "FromPort": 22,
        "ToPort": 22
    }
}
```

The following is example output.

```
{
    "NetworkInsightsPaths": {
        "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
        "NetworkInsightsPathArn": "arn:aws:ec2:us-east-1:123456789012:network-insights-
path/nip-0b26f224f1d131fa8",
        "CreatedDate": "2023-03-20T22:43:46.933Z",
        "Source": "igw-0797cccdc9d73b0e5",
        "Destination": "i-0495d385ad28331c7",
        "SourceArn": "arn:aws:ec2:us-east-1:123456789012:internet-
gateway/0797cccdc9d73b0e5",
        "DestinationArn": "arn:aws:ec2:us-east-1:123456789012:instance/0495d385ad28331c7",
        "Protocol": "tcp"
    }
}
```

To specify an IP address as the destination resource, omit the `--destination` parameter and filter on the destination address as follows.

```
aws ec2 create-network-insights-path
    --source igw-0797cccdc9d73b0e5
    --protocol TCP
    --filter-at-source file://source-filter.json
```

The following is an example of `source-filter.json`.

```
{
    "DestinationAddress": "34.230.71.227",
    "DestinationPortRange": {
        "FromPort": 22,
        "ToPort": 22
    }
}
```

# Step 2: Analyze the path

Use the following start-network-insights-analysis command to determine whether the destination is reachable using the protocol and port that you specified for the path. The analysis can take a few minutes to complete.

```
aws ec2 start-network-insights-analysis --network-insights-path-id nip-0b26f224f1d131fa8
```

The following is example output.

```
{
    "NetworkInsightsAnalysis": {
        "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a",
        "NetworkInsightsAnalysisArn": "arn:aws:ec2:us-east-1:123456789012:network-insights-
analysis/nia-02207aa13eb480c7a",
        "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
        "StartDate": "2023-03-20T22:58:37.495Z",
        "Status": "running"
    }
}
```

# Step 3: Get the results of the path analysis

After the path analysis completes, you can view the results using the describe-network-insights-analyses command.

```
aws ec2 describe-network-insights-analyses --network-insights-analysis-
ids nia-02207aa13eb480c7a
```

**Example 1: Not reachable**

The following is example output where the path is not reachable. When a path is not reachable, `NetworkPathFound` is `false` and `ExplanationCode` contains an explanation code. For descriptions of the explanation codes, see Reachability Analyzer explanation codes (p. 19). In this example,

ENI_SG_RULES_MISMATCH indicates that the security group does not allow the traffic. After you add a rule to the security group to allow the traffic, you can reanalyze the same path and confirm that it is reachable.


```
{
    "NetworkInsightsAnalyses": [
        {
            "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a",
            "NetworkInsightsAnalysisArn": "arn:aws:ec2:us-east-1:123456789012:network-
insights-analysis/nia-02207aa13eb480c7a",
            "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
            "StartDate": "2023-03-20T22:58:37.495Z",
            "Status": "succeeded",
            "NetworkPathFound": false,
            "ForwardPathComponents": [
                {
                    "SequenceNumber": 1,
                    "Component": {
                        "Id": "igw-0797cccdc9d73b0e5",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:internet-gateway/
igw-0797cccdc9d73b0e5"
                    },
                    "OutboundHeader": {
                        "DestinationAddresses": [
                            "10.0.4.120/32"
                        ]
                    },
                    "InboundHeader": {
                        "DestinationAddresses": [
                            "34.230.71.227/32"
                        ],
                        "DestinationPortRanges": [
                            {
                                "From": 22,
                                "To": 22
                            }
                        ],
                        "Protocol": "6",
                        "SourceAddresses": [
                            "11.0.0.0/32"
                        ],
                        "SourcePortRanges": [
                            {
                                "From": 0,
                                "To": 0
                            }
                        ]
                    },
                    "Vpc": {
                        "Id": "vpc-f1663d98ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-
f1663d98ad28331c7"
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                },
                {
                    "SequenceNumber": 2,
                    "AclRule": {
                        "Cidr": "0.0.0.0/0",
                        "Egress": "false",
                        "Protocol": "all",
                        "RuleAction": "allow",
                        "RuleNumber": 100
                    },
```

```
                    "Component": {
                        "Id": "acl-04fbcfb79260f6c5b",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:network-acl/
acl-04fbcfb79260f6c5b"
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                },
                {
                    "SequenceNumber": 3,
                    "AttachedTo": {
                        "Id": "i-0495d385ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:instance/
i-0495d385ad28331c7"
                    },
                    "Component": {
                        "Id": "eni-0a25edef15a6cc08c",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:network-interface/
eni-0a25edef15a6cc08c"
                    },
                    "Subnet": {
                        "Id": "subnet-004ff41eccb4d1194",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-004ff41eccb4d1194"
                    },
                    "Vpc": {
                        "Id": "vpc-f1663d98ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-
f1663d98ad28331c7"
                    },
                    "AdditionalDetails": [],
                    "Explanations": [
                        {
                            "Direction": "ingress",
                            "ExplanationCode": "ENI_SG_RULES_MISMATCH",
                            "NetworkInterface": {
                                "Id": "eni-0a25edef15a6cc08c",
                                "Arn": "arn:aws:ec2:us-east-1:123456789012:network-
interface/eni-0a25edef15a6cc08c"
                            },
                            "SecurityGroups": [
                                {
                                    "Id": "sg-02f0d35a850ba727f",
                                    "Arn": "arn:aws:ec2:us-east-1:123456789012:security-
group/sg-02f0d35a850ba727f"
                                }
                            ],
                            "Subnet": {
                                "Id": "subnet-004ff41eccb4d1194",
                                "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-004ff41eccb4d1194"
                            },
                            "Vpc": {
                                "Id": "vpc-f1663d98ad28331c7",
                                "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-
f1663d98ad28331c7"
                            },
                        }
                    ]
                },
                {
                    "SequenceNumber": 4,
                    "Component": {
                        "Id": "i-0495d385ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:instance/
i-0495d385ad28331c7"
```

```
                    },
                    "InboundHeader": {
                        "DestinationAddresses": [
                            "10.0.4.120/32"
                        ],
                        "DestinationPortRanges": [
                            {
                                "From": 22,
                                "To": 22
                            }
                        ],
                        "Protocol": "6",
                        "SourceAddresses": [
                            "11.0.0.0/32"
                        ],
                        "SourcePortRanges": [
                            {
                                "From": 0,
                                "To": 0
                            }
                        ]
                    },
                    "AdditionalDetails": [
                        {
                            "AdditionalDetailType": "UNIDIRECTIONAL_PATH_ANALYSIS_ONLY"
                        }
                    ],
                    "Explanations": []
                }
            ],
            "Explanations": [
                {
                    "Direction": "ingress",
                    "ExplanationCode": "ENI_SG_RULES_MISMATCH",
                    "NetworkInterface": {
                        "Id": "eni-0a25edef15a6cc08c",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:network-interface/
eni-0a25edef15a6cc08c"
                    },
                    "SecurityGroups": [
                        {
                            "Id": "sg-02f0d35a850ba727f",
                            "Arn": "arn:aws:ec2:us-east-1:123456789012:security-group/
sg-02f0d35a850ba727f"
                        }
                    ],
                    "Subnet": {
                        "Id": "subnet-004ff41eccb4d1194",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-004ff41eccb4d1194"
                    },
                    "Vpc": {
                        "Id": "vpc-f1663d98ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-
f1663d98ad28331c7"
                    }
                }
            ],
            "Tags": []
        }
    ]
}
```


**Example 2: Reachable**

The following is example output where the path is reachable. When a path is reachable, `NetworkPathFound` is `true`, `ForwardPathComponents` contains component-by-component details about the shortest reachable path from source to destination, and `ReturnPathComponents` contains component-by-component details about the shortest reachable path from destination to source.


```
{
    "NetworkInsightsAnalyses": [
        {
            "NetworkInsightsAnalysisId": "nia-076744f74a04c3c7f",
            "NetworkInsightsAnalysisArn": "arn:aws:ec2:us-east-1:123456789012:network-
insights-analysis/nia-076744f74a04c3c7f",
            "NetworkInsightsPathId": "nip-0614b9507b4e3e989",
            "StartDate": "2023-03-20T23:47:08.080Z",
            "Status": "succeeded",
            "NetworkPathFound": true,
            "ForwardPathComponents": [
                {
                    "SequenceNumber": 1,
                    "Component": {
                        "Id": "igw-0797cccdc9d73b0e5",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:internet-gateway/
igw-0797cccdc9d73b0e5",
                    },
                    "OutboundHeader": {
                        "DestinationAddresses": ["10.0.2.87/32"]
                    },
                    "InboundHeader": {
                        "DestinationAddresses": ["34.230.71.227/32"],
                        "DestinationPortRanges": [{
                            "From": 22,
                            "To": 22
                        }],
                        "Protocol": "6",
                        "SourceAddresses": ["0.0.0.0/5", "11.0.0.0/8", "12.0.0.0/6", ...],
                        "SourcePortRanges": [{
                            "From": 0,
                            "To": 65535
                        }]
                    },
                    "Vpc": {
                        "Id": "vpc-f1663d98ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-
f1663d98ad28331c7"
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                },
                {
                    "SequenceNumber": 2,
                    "AclRule": {
                        "Cidr": "0.0.0.0/0",
                        "Egress": false,
                        "Protocol": "all",
                        "RuleAction": "allow",
                        "RuleNumber": 100
                    },
                    "Component": {
                        "Id": "acl-04fbcfb79260f6c5b",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:network-acl/
acl-04fbcfb79260f6c5b"
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                },
                {
```

```
                    "SequenceNumber": 3,
                    "Component": {
                        "Id": "sg-02f0d35a850ba727f",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:security-group/
sg-02f0d35a850ba727f"
                    },
                    "SecurityGroupRule": {
                        "Cidr": "0.0.0.0/0",
                        "Direction": "ingress",
                        "PortRange": {
                            "From": 22,
                            "To": 22
                        },
                        "Protocol": "tcp"
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                },
                {
                    "SequenceNumber": 4,
                    "AttachedTo": {
                        "Id": "i-0495d385ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:instance/
i-0495d385ad28331c7"
                    },
                    "Component": {
                        "Id": "eni-0a25edef15a6cc08c",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:network-interface/
eni-0a25edef15a6cc08c"
                    },
                    "Subnet": {
                        "Id": "subnet-004ff41eccb4d1194",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-004ff41eccb4d1194"
                    },
                    "Vpc": {
                        "Id": "vpc-f1663d98ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-
f1663d98ad28331c7"
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                },
                {
                    "SequenceNumber": 5,
                    "Component": {
                        "Id": "i-0626d4edd54f1286d",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:instance/
i-0626d4edd54f1286d"
                    },
                    "InboundHeader": {
                        "DestinationAddresses": ["10.0.4.120/32"],
                        "DestinationPortRanges": [{
                            "From": 22,
                            "To": 22
                        }],
                        "Protocol": "6",
                        "SourceAddresses": ["0.0.0.0/5", "11.0.0.0/8", "12.0.0.0/6", ...],
                        "SourcePortRanges": [{
                            "From": 0,
                            "To": 65535
                        }]
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                }
```

```
            ],
            "ReturnPathComponents": [
                {
                    "SequenceNumber": 1,
                    "Component": {
                        "Id": "i-0626d4edd54f1286d",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:instance/
i-0626d4edd54f1286d"
                    },
                    "OutboundHeader": {
                        "DestinationAddresses": ["0.0.0.0/5", "11.0.0.0/8",
 "12.0.0.0/6", ...],
                        "DestinationPortRanges": [{
                            "From": 0,
                            "To": 65535
                        }],
                        "Protocol": "6",
                        "SourceAddresses": ["10.0.2.87/32"],
                        "SourcePortRanges": [{
                            "From": 22,
                            "To": 22
                        }]
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                },
                {
                    "SequenceNumber": 2,
                    "AttachedTo": {
                        "Id": "i-0495d385ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:instance/
i-0495d385ad28331c7"
                    },
                    "Component": {
                        "Id": "eni-0a25edef15a6cc08c",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:network-interface/
eni-0a25edef15a6cc08c"
                    },
                    "Subnet": {
                        "Id": "subnet-004ff41eccb4d1194",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-004ff41eccb4d1194"
                    },
                    "Vpc": {
                        "Id": "vpc-f1663d98ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-
f1663d98ad28331c7"
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                },
                {
                    "SequenceNumber": 3,
                    "Component": {
                        "Id": "sg-02f0d35a850ba727f",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:security-group/
sg-02f0d35a850ba727f"
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                },
                {
                    "SequenceNumber": 4,
                    "AclRule": {
                        "Cidr": "0.0.0.0/0",
                        "Egress": true,
```

```
                        "Protocol": "all",
                        "RuleAction": "allow",
                        "RuleNumber": 100
                    },
                    "Component": {
                        "Id": "acl-0a8e20a0a9f144d36",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:network-acl/
acl-0a8e20a0a9f144d36"
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                },
                {
                    "SequenceNumber": 5,
                    "Component": {
                        "Id": "rtb-0d49a54c0a8c0bd9b",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:route-table/
rtb-0d49a54c0a8c0bd9b"
                    },
                    "RouteTableRoute": {
                        "DestinationCidr": "0.0.0.0/0",
                        "GatewayId": "igw-0797cccdc9d73b0e5",
                        "Origin": "createroute",
                        "State": "active"
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                },
                {
                    "SequenceNumber": 6,
                    "Component": {
                        "Id": "igw-0797cccdc9d73b0e5",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:internet-gateway/
igw-0797cccdc9d73b0e5"
                    },
                    "OutboundHeader": {
                        "DestinationAddresses": ["0.0.0.0/5", "11.0.0.0/8",
 "12.0.0.0/6", ...],
                        "DestinationPortRanges": [{
                            "From": 0,
                            "To": 65535
                        }],
                        "Protocol": "6",
                        "SourceAddresses": ["34.230.71.227/32"],
                        "SourcePortRanges": [{
                            "From": 22,
                            "To": 22
                        }]
                    },
                    "Vpc": {
                        "Id": "vpc-f1663d98ad28331c7",
                        "Arn": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-
f1663d98ad28331c7"
                    },
                    "AdditionalDetails": [],
                    "Explanations": []
                }
            ],
            "Tags": []
        }
    ]
}
```

# Step 4: Delete the path

If you no longer need the path, you can delete it. Before you can delete the path, you must delete its analyses.

**To delete the path**

1. Use the following delete-network-insights-analysis command to delete the path analysis.

```
aws ec2 delete-network-insights-analysis --network-insights-analysis-
id nia-02207aa13eb480c7a
```

2. Use the following delete-network-insights-path to delete the path.

```
aws ec2 delete-network-insights-path --network-insights-path-id nip-0b26f224f1d131fa8
```

If you keep the path, note that Reachability Analyzer will automatically delete the analysis 120 days after its creation date.

# Reachability Analyzer explanation codes

If a destination is not reachable, Reachability Analyzer provides one or more explanation codes to help you diagnose and address network misconfiguration.

**Contents**



## Path is not reachable

The following explanation codes indicate that the path analysis determined that the path is not reachable.

**BAD_STATE**

This component is not in a functional state.

**BAD_STATE_ATTACHMENT**

The attachment between these components is not in a functional state.

**BAD_STATE_ROUTE**

This route is not in a functional state.

**BAD_STATE_VPN**

This VPN connection is not in a functional state.

**CANNOT_ROUTE**

This route can't transmit traffic because its destination CIDR or prefix list does not match the destination address of the packet.

**ELB_ACL_RESTRICTION**

Classic Load Balancers apply network ACLs to outbound traffic, even if it's destined for a target in the same subnet as the load balancer.

**ELB_INSTALLED_AZ_RESTRICTION**

This load balancer can send traffic only to targets in Availability Zones that are enabled for the load balancer.

**ELB_LISTENER_PORT_RESTRICTION**

This Classic Load Balancer listener allows only inbound traffic destined for the specified port, and outbound traffic with the specified destination port.

**ELB_LISTENERS_MISMATCH**

This Classic Load Balancer does not have a listener that accepts the traffic.

**ELB_NOT_CROSSZONE**

This load balancer can't send traffic to some targets because cross-zone load balancing is disabled.

**ELBV2_LISTENER_HAS_NO_TG**

This listener is associated with target groups that have no targets.

**ELBV2_LISTENER_PORT_RESTRICTION**

This listener does not accept traffic unless it has the specified destination port.

**ELBV2_LISTENER_REQUIRES_TG_ACCEPT**

This listener does not have a target group that accepts the traffic.

**ELBV2_LISTENERS_MISMATCH**

This load balancer does not have a listener that accepts the traffic.

**ELBV2_NO_TARGETS_IN_AZ**

The load balancer does not have targets in the specified Availability Zones.

**ELBV2_SOURCE_ADDRESS_PRESERVATION**

If source address preservation is enabled, the outgoing source address is unaltered while traversing the Network Load Balancer.

**ENI_ADDRESS_RESTRICTION**

This network interface does not allow inbound or outbound traffic unless the source or destination address matches its private IP address.

**ENI_SG_RULES_MISMATCH**

This security group has no inbound or outbound rules that apply.

**ENI_SOURCE_DEST_CHECK_RESTRICTION**

Network interfaces with source/destination check enabled reject inbound traffic if the destination address does not match one of its private IP addresses, and reject outbound traffic if the source address does not match one of their private IP addresses.

**FIREWALL_RULES_RESTRICTION**

The traffic is blocked by a matching Network Firewall firewall rule.

**GATEWAY_REJECTS_SPOOFED_TRAFFIC**

Gateways reject traffic from network interfaces if the source IP address is not a public IP address associated with the network interface.

**GWLB_DESTINATION_PORT_RESTRICTION**

Traffic between a Gateway Load Balancer and its targets must use port 6081 as the destination port. To analyze connectivity through a Gateway Load Balancer, specify port 6081 in the path definition.

**GWLB_PROTOCOL_RESTRICTION**

Traffic between a Gateway Load Balancer and its targets must use the GENEVE protocol, which is UDP-based. To analyze connectivity through a Gateway Load Balancer, specify the UDP protocol in the path definition.

**HIGHER_PRIORITY_ROUTE**

This route table contains a route to the destination that can't be used because there is a higher priority route with the same destination CIDR.

**IGW_DESTINATION_ADDRESS_IN_VPC_CIDRS**

Internet gateways accept traffic only if the destination address is within the VPC CIDR block.

**IGW_DESTINATION_ADDRESS_NOT_IN_RFC1918_EGRESS**

Internet gateways reject outbound traffic with destination addresses in the private IP address range (see RFC1918).

**IGW_DESTINATION_ADDRESS_NOT_IN_RFC6598_EGRESS**

Internet gateways reject outbound traffic with destination addresses in the shared IP address range (see RFC6598).

**IGW_NAT_REFLECTION**

The path has an internet gateway as an intermediate component, which Reachability Analyzer does not support. Instead, analyze the path from the source to the internet gateway and then analyze the path from the internet gateway to the destination.

**IGW_PRIVATE_IP_ASSOCIATION_FOR_INGRESS**

Internet gateways reject inbound traffic with a destination address that is not the public IP address of a network interface in the VPC with an available attachment.

**IGW_PUBLIC_IP_ASSOCIATION_FOR_EGRESS**

Traffic can't reach the internet through the internet gateway if the source address is not paired with a public IP address or if the source address does not belong to a network interface in the VPC with an available attachment.

**IGW_SOURCE_ADDRESS_NOT_IN_RFC1918_INGRESS**

Internet gateways reject inbound traffic with source addresses in the private IP address range (see RFC1918).

**IGW_SOURCE_ADDRESS_NOT_IN_RFC6598_INGRESS**

Internet gateways reject inbound traffic with source addresses in the shared IP address range (see RFC6598).

**INGRESS_RTB_NO_PUBLIC_IP**

A middlebox appliance can't receive traffic from the internet through an ingress route table if it does not have a public IP address.

**INGRESS_RTB_TRAFFIC_REDIRECTION**

Subnets whose traffic is redirected to a middlebox appliance can't use a direct route to the internet gateway even when the subnet route table provides one.

**MORE_SPECIFIC_ROUTE**

The specified route can't be used to transmit traffic because there is a more specific route that matches. You can use filters to require that a path include a specific intermediate component.

**NGW_DEST_ADDRESS_PRESERVATION**

NAT gateways do not alter destination addresses.

**NGW_REQUIRES_SOURCE_IN_VPC**

NAT gateways can only transmit traffic that originates from network interfaces within the same VPC. NAT gateways can't transmit traffic that originates from peering connections, VPN connections, or AWS Direct Connect.

**NGW_SOURCE_ADDRESS_REASSIGN**

NAT gateways transform the source's addresses in outbound traffic to match its private IP address.

**NO_POSSIBLE_DESTINATION**

The network component can't deliver the packet to any possible destination, or the network component sent traffic to a destination in another account or Region. If the destination is in another account, enable cross-account analyses (p. 28).

**NO_ROUTE_TO_DESTINATION**

The route table does not have an applicable route to the destination resource.

**PCX_REQUIRES_ADDRESS_IN_VPC_CIDR**

Traffic can traverse this peering connection only if the destination or source address is within the CIDR block of the destination VPC.

**PROTOCOL_RESTRICTION**

This component only accepts traffic with specific protocols.

**REMAP_EPHEMERAL_PORT**

Outbound traffic from a NAT gateway or load balancer has the source port remapped to an ephemeral port in the range [1024–65535].

**SG_HAS_NO_RULES**

This security group has no inbound or outbound rules.

**SUBNET_ACL_RESTRICTION**

Inbound or outbound traffic for a subnet must be admitted by the network ACL for the subnet.

**TARGET_ADDRESS_RESTRICTION**

A load balancer can only route traffic that is destined for the address of one of its targets.

**TARGET_PORT_RESTRICTION**

A load balancer can only route traffic to a target using its registered port.

**TGW_ATTACH_MISSING_TGW_RTB_ASSOCIATION**

This transit gateway attachment doesn't have a valid transit gateway route table association.

**TGW_ATTACH_VPC_AZ_RESTRICTION**

Traffic from a VPC attachment in the default mode can't be forwarded to the network interface in this Availability Zone because it comes from an Availability Zone where the attachment has a different network interface. Traffic from a VPC attachment in appliance mode can't be forwarded to the network interface in this Availability Zone because on the forward path it used a different Availability Zone.

**TGW_BAD_STATE_VPN**

This VPN connection is in a non-functional state.

**TGW_ROUTE_AZ_RESTRICTION**

This transit gateway is not registered in the Availability Zone where the traffic originates. The VPC attachment must have a subnet association in the Availability Zone.

**TGW_RTB_BAD_STATE_ROUTE**

This transit gateway route table has a route to the destination that is in a bad state.

**TGW_RTB_CANNOT_ROUTE**

This transit gateway route table has a route to the intended destination, but the route does not match the package destination address.

**TGW_RTB_HIGHER_PRIORITY_ROUTE**

This transit gateway route table contains a route to the intended destination that can't be used because there is a higher-priority route with the same destination CIDR.

**TGW_RTB_MORE_SPECIFIC_ROUTE**

This transit gateway route table has a route to the destination, but there is a more specific route.

**TGW_RTB_NO_ROUTE_TO_TGW_ATTACHMENT**

This transit gateway route table has no route to this transit gateway attachment.

**TGW_RTB_ROUTES_ARE_UNKNOWN**

The routes of this transit gateway route table are not known. This might be due to an internal error or because the transit gateway route table does not belong to the account running the analysis.

**UNKNOWN_DESTINATION**

The path can't be extended because the information about the destination is insufficient.

**UNKNOWN_PEERED_SGS**

One of the VPCs in the VPC peering connection is unknown. This is typically because the VPC is in a different account. Access controls referencing security groups are treated as inaccessible and deny traffic crossing this peering connection.

**VGW_PRIVATE_IP_ASSOCIATION_FOR_EGRESS**

Virtual private gateways can't accept outbound traffic if the source address does not belong to a network interface in the VPC with an available attachment.

**VGW_PRIVATE_IP_ASSOCIATION_FOR_INGRESS**

Virtual private gateways can't accept inbound traffic if the destination address is not the private IP address of a network interface in the VPC with an available attachment.

**VPC_LOCAL_ROUTE_CIDR_RESTRICTION**

Local routes apply only to packets with a destination address within the VPC CIDR block.

**VPCE_GATEWAY_EGRESS_SOURCE_ADDRESS_RESTRICTION**

VPC gateway endpoints emit only traffic with source addresses within the CIDRs of their corresponding prefix lists.

**VPCE_GATEWAY_PROTOCOL_RESTRICTION**

VPC gateway endpoints accept only TCP or ICMP ECHO traffic, and emit only TCP or ICMP ECHO reply traffic.

**VPCE_SERVICE_NOT_INSTALLED_IN_AZ**

The VPC endpoint service is not installed in the specified Availability Zone.

# Configuration

The following explanation codes indicate that the path analysis determined that no path is possible.

**DISCONNECTED_VPCS**

The source and destination are in separate VPCs that are not connected by a supported resource.

**NO_PATH**

Reachability Analyzer was unable to find a path from the source to the destination. The following are the most common causes:

- The path does not meet the optional configuration details, such as an IP address, port, or filter.
- The source or destination components are temporarily isolated from the network (for example, a newly started instance that does not yet have a network interface).
- The source can't initiate traffic to the destination (for example, an interface VPC endpoint or gateway VPC endpoint can't initiate connections with components in the same VPC as the VPC endpoint).
- The path requires the ability to analyze an unsupported feature (for example, IPv6) or an unsupported network component.

**NO_SOURCE_OR_DESTINATION**

The source or destination resource does not exist.

**UNASSOCIATED_COMPONENT**

The component is not associated with a VPC in your account (for example, a recently terminated instance), or none of its network interfaces has an IPv4 address.

**UNSUPPORTED_COMPONENT**

The component is not supported by Reachability Analyzer.

# Search filter codes

The following explanation codes indicate that the path analysis couldn't find a path from the source to the destination that matched the specified filters. However, there might be a path that matches some of the specified filters. Verify that the filters are as intended. Otherwise, remove the filters that didn't match.

**COMPONENT_FILTER_RESTRICTION**

There is no path that traverses the specified component.

**COMPONENT_FILTER_RESTRICTION_REMOVED_COMPONENT**

There is no path that traverses the specified component because of an intermediate component filter.

**FILTER_AT_DESTINATION_DESTINATION_ADDRESS**

There is no path that matches the specified destination IP address at the destination.

**FILTER_AT_DESTINATION_DESTINATION_PORT_RANGE**

There is no path that matches the specified destination port range at the destination.

**FILTER_AT_DESTINATION_SOURCE_ADDRESS**

There is no path that matches the specified source address at the destination.

**FILTER_AT_DESTINATION_SOURCE_PORT_RANGE**

There is no path that matches the specified source port range at the destination.

**FILTER_AT_SOURCE_DESTINATION_ADDRESS**

There is no path that matches the specified destination IP address at the source.

**FILTER_AT_SOURCE_DESTINATION_PORT_RANGE**

There is no path that matches the specified destination port range at the source.

**FILTER_AT_SOURCE_PROTOCOL**

There is no path that matches the specified protocol.

**FILTER_AT_SOURCE_SOURCE_ADDRESS**

There is no path that matches the specified source IP address at the source.

**FILTER_AT_SOURCE_SOURCE_PORT_RANGE**

There is no path that matches the specified source port range at the source.

**IGW_EXPECTS_PUBLIC_ADDRESS**

IP addresses must be public IP addresses when the resource is an internet gateway.

# Reachability Analyzer additional detail codes

Reachability Analyzer uses additional detail codes to provide information about the result of a path analysis.

The following additional detail codes are supported.

**ASSUMPTION_PRESERVE_CLIENT_IP_IS_DISABLED**

The analysis could not describe target group attributes for the target group, so the network path is based on the assumption that client IP preservation is disabled on the target group. You should verify this assumption.

**ASSUMPTION_PRESERVE_CLIENT_IP_IS_ENABLED**

The analysis could not describe target group attributes for the target group, so the network path is based on the assumption that client IP preservation is enabled on the target group. You should verify this assumption.

**AVAILABILITY_ZONE_CROSSED**

The network path crosses Availability Zones.

**FIREWALL_UNSUPPORTED_HIGHER_PRIORITY_RULE_GROUP_TYPE**

There is at least one higher priority rule that could match the traffic in this path, but we ignored because it contains an unsupported rule type. Verify that the result of the analysis matches the behavior of AWS Network Firewall in your network.

**FIREWALL_UNSUPPORTED_HIGHER_PRIORITY_RULES**

There is at least one higher priority rule that could match the traffic in this path, but we ignored because it contains an unsupported rule option. Verify that the result of the analysis matches the behavior of AWS Network Firewall in your network.

**FIREWALL_UNSUPPORTED_RULE_OPTIONS**

The matching firewall rule contains an unsupported rule option. Verify that the result of the analysis matches the behavior of AWS Network Firewall in your network.

**MISSING_ROUTE_TO_TRANSIT_GATEWAY**

Traffic is routed from the transit gateway to the VPC endpoint. However, there is no route from the VPC endpoint to the transit gateway, so we might drop the response traffic.

**MISSING_TARGET_GROUP_ATTRIBUTES**

The target group attributes for the target were missing, so the analysis could not consider them.

**PATH_THROUGH_GWLB_NOT_CHECKED**

The analysis does not consider that traffic entering the VPC endpoint is forwarded to a Gateway Load Balancer for inspection before exiting the VPC endpoint.

**RECOMMENDED_APPLIANCE_MODE**

The transit gateway VPC attachment has appliance mode disabled, but traffic is inspected through a Network Firewall. We recommend that you enable appliance mode for the VPC attachment.

**UNIDIRECTIONAL_PATH_ANALYSIS_ONLY**

The results include forward path analysis from the source to the destination. There might be a blocking configuration in the reverse path, which could not be analyzed.

# Cross-account analyses for Reachability Analyzer

Reachability Analyzer analyzes the path between a source and destination. To analyze paths across multiple AWS accounts, enable trusted access for Reachability Analyzer with your organization from AWS Organizations. You can also register member accounts as delegated administrator accounts. A user in the management account or a delegated administrator account can define a path and run an analysis using a source from any account in the organization, and a destination resource from any account in the organization.

There is no additional charge to run cross-account analyses.

**Considerations**

- Before accounts in the organization can use this feature in an opt-in Region, the management account must enable the opt-in Region. For more information, see Enabling a Region in the *AWS General Reference*.

**Tasks**



## Enable trusted access

When you enable trusted access, Reachability Analyzer deploys the AWSServiceRoleForReachabilityAnalyzer (p. 41) service-linked role and the required cross-account access roles (p. 44) to all accounts in your organization.

**To enable trusted access using the console**

1. Sign in to the management account.
2. Open the Network Manager console at https://console.aws.amazon.com/networkmanager/home.
3. From the navigation pane, choose **Reachability Analyzer**, **Settings**.
4. For **Trusted Access**, choose **Turn on trusted access**.
5. Do not close or navigate away from this page until you see a success notification indicating that trusted access is turned on. This can take several minutes.

**To enable trusted access using the AWS CLI**

From the management account, use the enable-reachability-analyzer-organization-sharing command.

## IAM role deployments

When you enable trusted access, the following roles are deployed in your organization:

- AWSServiceRoleForReachabilityAnalyzer (p. 42) – The service-linked role for Reachability Analyzer.
- IAMRoleForReachabilityAnalyzerCrossAccountResourceAccess (p. 44) – The role for cross-account resource access for Reachability Analyzer.
- AWSServiceRoleForCloudFormationStackSetsOrgAdmin – The service-linked role for AWS CloudFormation StackSets for the management account.
- AWSServiceRoleForCloudFormationStackSetsOrgMember – The service-linked role for AWS CloudFormation StackSets for the member accounts.

The deployments can take several minutes to complete, depending on the number of member accounts in your organization. You can view the status of the role deployments as follows.

**To view IAM role deployments**

1. Sign in to the management account.
2. Open the Network Manager console at https://console.aws.amazon.com/networkmanager/home.
3. From the navigation pane, choose **Reachability Analyzer**, **Settings**.
4. Check **IAM role deployments status**.

# Manage delegated administrator accounts

You can register up to 5 delegated administrator accounts. If you deregister a delegated administrator account, the users in the account can't run a new cross-account analysis, but they can still see the previously run analyses.

**To manage delegated administrators**

1. Sign in to the management account.
2. Open the Network Manager console at https://console.aws.amazon.com/networkmanager/home.
3. From the navigation pane, choose **Reachability Analyzer**, **Settings**.
4. To register a member account as a delegated administrator account, choose **Register delegated administrator**. Select the check box for the account, and then choose **Register delegated administrator**.
5. To deregister a delegated administrator account, select the checkbox for the account, and then choose **Deregister**.

# Disable trusted access

After you disable trusted access, the users in the management account and delegated administrator accounts can't run a new cross-account analysis. However, they can still see the previously run analyses. Before you can disable trusted access, you must deregister the delegated administrator accounts.

You can enable trusted access again after disabling it. However, you must first re-register the delegated administrator accounts.

**To disable trusted access using the console**

1. Sign in to the management account.
2. Open the Network Manager console at https://console.aws.amazon.com/networkmanager/home.
3. From the navigation pane, choose **Reachability Analyzer**, **Settings**.
4. For **Trusted Access**, choose **Turn off trusted access**.

5. Do not close or navigate away from this page until you see a success notification indicating that trusted access is turned off. This can take several minutes.

**To disable trusted access using the AWS CLI**

From the management account, use the disable-aws-service-access command.

# Troubleshoot

The following information can help you troubleshoot common issues.

**Issues**



## "StackSet is not empty" or "StackSet already exists"

If you receive one of these errors while enabling trusted access, do the following to resolve the issue.

**To resolve the issue**

1. Choose **Turn off trusted access**.
2. Wait until you see a banner at the top of the screen indicating that the operation was successful.
3. Choose **Turn on trusted access**.

## "Organizational unit not found in StackSet"

If you receive this error while disabling trusted access, do the following to resolve the issue.

**To resolve the issue**

1. Open the CloudFront console at https://console.aws.amazon.com/cloudfront/v3/home.
2. In the navigation pane, choose **StackSets**.
3. Select `ReachabilityAnalyzerCrossAccountResourceAccessStackSet` and then choose **Actions**, **Delete StackSet**.
4. Return to the Reachability Analyzer settings page and refresh the page.
5. Choose **Turn off trusted access**.

# Identity and access management for Reachability 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 Reachability Analyzer resources. IAM is an AWS service that you can use with no additional charge.

**Contents**



## Audience

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

**Service user** – If you use the Reachability Analyzer service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Reachability 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 Reachability Analyzer resources at your company, you probably have full access to Reachability Analyzer. It's your job to determine which Reachability 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 Reachability Analyzer.

## Authenticating with identities

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

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (successor to AWS Single Sign-On) (IAM Identity Center) users, your company's

single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

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

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

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide* and Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

# AWS account root user

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

# Federated identity

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

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

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

# IAM users and groups

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

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

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

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

# IAM roles

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

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

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, resources, and condition keys for Amazon EC2 in the *Service Authorization Reference*.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the

EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

# Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

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

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

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

## Identity-based policies

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

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

## Resource-based policies

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

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

## Access control lists (ACLs)

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

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

## Other policy types

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

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

## Multiple policy types

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

# How Reachability Analyzer works with IAM

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

**IAM features you can use with Reachability Analyzer**

| IAM feature | Reachability Analyzer support |
|---|---|
| Identity-based policies (p. 36) | Yes |
| Resource-based policies (p. 36) | No |
| Policy actions (p. 37) | Yes |
| Policy resources (p. 37) | Yes |
| Policy condition keys (service-specific) (p. 38) | No |
| ACLs (p. 38) | No |

| IAM feature | Reachability Analyzer support |
|---|---|
| ABAC (tags in policies) (p. 38) | Yes |
| Temporary credentials (p. 39) | Yes |
| Principal permissions (p. 39) | Yes |
| Service roles (p. 40) | No |
| Service-linked roles (p. 40) | Yes |

To get a high-level view of how AWS FIS 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 Reachability 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 Creating IAM policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

# Resource-based policies within Reachability 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. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

# Policy actions for Reachability 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. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

Reachability Analyzer shares its API namespace with Amazon EC2. Policy actions in Reachability 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 Reachability Analyzer:

- `CreateNetworkInsightsPath`
- `DeleteNetworkInsightsAnalysis`
- `DeleteNetworkInsightsPath`
- `DescribeNetworkInsightsAnalyses`
- `DescribeNetworkInsightsPaths`
- `EnableReachabilityAnalyzerOrganizationSharing`
- `StartNetworkInsightsAnalysis`

For more information, see Actions Defined by Amazon EC2 in the *Service Authorization Reference*.

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

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

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

The following Reachability Analyzer API actions do not support resource-level permissions.

- `DescribeNetworkInsightsAnalyses`
- `DescribeNetworkInsightsPaths`

# Policy condition keys for Reachability 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 (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

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

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

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

# ACLs in Reachability 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 Reachability Analyzer

| | |
|---|---|
| Supports ABAC (tags in policies) | Yes |

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

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

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

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

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

# Using temporary credentials with Reachability Analyzer

| | |
|---|---|
| Supports temporary credentials | Yes |

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

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

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

# Cross-service principal permissions for Reachability Analyzer

| | |
|---|---|
| Supports principal permissions | Yes |

When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, resources, and condition keys for Amazon EC2 in the *Service Authorization Reference*.

## Service roles for Reachability 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 Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

## Service-linked roles for Reachability Analyzer

| | |
|---|---|
| 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.

For details about creating or managing Reachability Analyzer service-linked roles, see Use service-linked roles for Reachability Analyzer (p. 41).

# Required API permissions for Reachability Analyzer

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

- AWS Direct Connect
- Amazon EC2
- Elastic Load Balancing
- AWS Global Accelerator
- AWS Network Firewall
- AWS Tiros

To view the permissions for this policy, see AmazonVPCReachabilityAnalyzerFullAccessPolicy in the *AWS Managed Policy Reference*.

## Additional information

**Reachability Analyzer API calls**

The following permissions are required to call the Reachability Analyzer APIs. Users need these permissions to create and start analyzing a specified path for reachability, or to view and delete existing paths and analyses in your account. You must grant users permission to call the Reachability Analyzer API actions they need.

- `ec2:CreateNetworkInsightsPath`
- `ec2:DeleteNetworkInsightsAnalysis`
- `ec2:DeleteNetworkInsightsPath`
- `ec2:DescribeNetworkInsightsAnalyses`

- `ec2:DescribeNetworkInsightsPaths`
- `ec2:EnableReachabilityAnalyzerOrganizationSharing`
- `ec2:StartNetworkInsightsAnalysis`

#### Describe API calls for networking-related resources

Reachability Analyzer uses describe API calls while gathering information about your resources from Amazon VPC, Amazon EC2, and Elastic Load Balancing (for example, subnets, network interfaces, and security groups). To access Reachability Analyzer, users must also have these API permissions.

### Cross-account analysis

The following permissions are required to establish a trust relationship between Reachability Analyzer and AWS Organizations. After you establish a trust relationship, a user in the management account or a delegated administrator account can run cross-account analyses using resources from the member accounts.

- `cloudformation:EnableOrganizationsAccess`
- `iam:CreateServiceLinkedRole`
- `iam:GetRole`
- `organizations:EnableAWSServiceAccess`
- `organizations:DescribeOrganization`
- `organizations:DisableAWSServiceAccess`
- `organizations:ListRoots`

### Tagging-related API calls

To tag or untag Reachability 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 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 reachability findings to Reachability Analyzer. Calls to the Tiros endpoint are required for Reachability Analyzer to function. To access Reachability Analyzer, users must also have the same API permissions.

# Use service-linked roles for Reachability Analyzer

Reachability Analyzer uses AWS Identity and Access Management (IAM) service-linked roles for multi-account analysis. A service-linked role is a unique type of IAM role that is linked directly to Reachability Analyzer. Service-linked roles are predefined by Reachability Analyzer and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Reachability Analyzer easier because you don't have to add the necessary permissions yourself. Reachability Analyzer defines the permissions of its service-linked roles, and unless defined otherwise, only Reachability Analyzer can assume its roles. The defined permissions

include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

# Service-linked role permissions for Reachability Analyzer

Reachability Analyzer uses the service-linked role named **AWSServiceRoleForReachabilityAnalyzer** to access AWS resources and integrate with AWS Organizations on your behalf.

The **AWSServiceRoleForReachabilityAnalyzer** role trusts the following services to assume the role:

- `reachabilityanalyzer.networkinsights.amazonaws.com`

The **AWSServiceRoleForReachabilityAnalyzer** service-linked role uses the managed policy ??? (p. 43).

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*.

# Create a service-linked role for Reachability Analyzer

You don't need to create this service-linked role yourself. When you enable integration with AWS Organizations, Reachability Analyzer creates the **AWSServiceRoleForReachabilityAnalyzer** role for you. For more information, see the section called “Enable trusted access” (p. 28).

If you delete this service-linked role and then enable integration with AWS Organizations, Reachability Analyzer creates the **AWSServiceRoleForReachabilityAnalyzer** role for you again.

# Edit a service-linked role for Reachability Analyzer

Reachability Analyzer does not allow you to edit the **AWSServiceRoleForReachabilityAnalyzer** role. After you create a service-linked role, you cannot 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*.

# Delete a service-linked role for Reachability Analyzer

If you are finished performing multi-account analysis, we recommend that you delete the **AWSServiceRoleForReachabilityAnalyzer** role. You can delete this service-linked role only after you disable the integration of Reachability Analyzer with AWS Organizations.

If the Reachability Analyzer service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

**To disable integration with AWS Organizations**

Make sure that you are not running a path analysis. To disable integration using the Reachability Analyzer console, see the section called “Disable trusted access” (p. 29). To disable integration using the AWS CLI or an API, see How to enable or disabled trusted access in the *AWS Organizations User Guide*.

**To delete the service-linked role using IAM**

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

# AWS managed policies for Reachability 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: AmazonVPCReachabilityAnalyzerFullAccessPolicy

Provides permissions to create, analyze, and delete paths, and to describe path resources, such as EC2 instances, firewalls, internet gateways, load balancers, NAT gateways, network interfaces, transit gateways, VPC endpoint services, VPC endpoints, VPC peering connections, and virtual private gateways.

To view the permissions for this policy, see AmazonVPCReachabilityAnalyzerFullAccessPolicy in the *AWS Managed Policy Reference*.

## AWS managed policy: AmazonVPCReachabilityAnalyzerPathComponentReadPolicy

This policy is attached to the role the section called "IAMRoleForReachabilityAnalyzerCrossAccountResourceAccess" (p. 44). This role is deployed to the member accounts in an organization when the management account enables trusted access for Reachability Analyzer using the console. It provides permissions to view resources from across your organization using the Reachability Analyzer console. For more information, see Cross-account access roles (p. 44).

To view the permissions for this policy, see AmazonVPCReachabilityAnalyzerPathComponentReadPolicy in the *AWS Managed Policy Reference*.

## AWS managed policy: AWSReachabilityAnalyzerServiceRolePolicy

This policy is attached to a service-linked role that allows Reachability Analyzer to perform actions on your behalf. For more information, see Use service-linked roles (p. 41).

To view the permissions for this policy, see AWSReachabilityAnalyzerServiceRolePolicy in the *AWS Managed Policy Reference*.

## Reachability Analyzer updates to AWS managed policies

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

| Change | Description | Date |
| --- | --- | --- |
| AmazonVPCReachabilityAnalyzerFullAccess policy (p. 47) – New policy | Added a policy that provides full access to Reachability Analyzer for single account use. | June 14, 2023 |
| AmazonVPCReachabilityAnalyzerPathComponentReadOnly policy (p. 47) – New policy | Added a policy that grants member accounts permission to view resources from across your organization. The policy is attached to a role that is deployed to member accounts when the management account enables trusted access for Reachability Analyzer using the console. | May 1, 2023 |
| AWSReachabilityAnalyzerServiceRolePolicy (p. 47) - New policy | Added a policy that is attached to a service-linked role that allows it to access AWS resources and integrate with AWS Organizations on your behalf. | November, 23, 2022 |
| Reachability Analyzer started tracking changes | Reachability Analyzer started tracking changes for its AWS managed policies. | March 1, 2021 |

# Cross-account access roles for Reachability Analyzer

When you enable trusted access for Reachability Analyzer, we use AWS CloudFormation StackSets to deploy the IAMRoleForReachabilityAnalyzerCrossAccountResourceAccess IAM role to all member accounts in the organization. This role allows the management account and delegated administrator accounts to specify resources from member accounts in path analyses.

Reachability Analyzer creates the custom IAM role automatically when you turn on trusted access using the Network Manager console. We strongly recommend that you use the console to turn on trusted access, as alternate approaches require an advanced level of expertise and are more prone to error.

Deregistering a delegated administrator removes it from the account list so that it can no longer assume this custom IAM role. If you turn off trusted access, we delete the StackSets.

## IAMRoleForReachabilityAnalyzerCrossAccountResourceAcces

This IAM policy role enables cross-account read-only access to resources through role switching. For more information, see AmazonEC2ReadOnlyAccess and AWSDirectConnectReadOnlyAccess in the IAM console.

```
AWSTemplateFormatVersion: '2010-09-09'
Description: Enables Console Access role
Resources:
```

```
ConsoleRole:
  Type: AWS::IAM::Role
  Properties:
    RoleName: IAMRoleForReachabilityAnalyzerCrossAccountResourceAccess
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
      - Effect: Allow
        Principal:
          AWS:
          - arn:aws:iam::management-account-id:root
          - arn:aws:iam::delegated-admin-1-account-id:root
          - arn:aws:iam::delegated-admin-2-account-id:root
        Action:
        - sts:AssumeRole
    Path: "/"
    ManagedPolicyArns:
    - arn:aws::iam::aws:policy/AWSDirectConnectReadOnlyAccess
    - arn:aws::iam::aws:policy/AmazonEC2ReadOnlyAccess
    - arn:aws::iam::aws:policy/AmazonVPCReachabilityAnalyzerPathComponentReadPolicy
```

# Manage IAM role deployments

If you make changes to your role policies, or if you've updated a self-managed role, you can deploy the updated policy to the accounts in your organization.

With a self-managed deployment, you are responsible for attaching the required policies and managing the trust relationship required for the delegated administrator and management accounts to use cross-account analyses.

# Troubleshoot self-managed role deployments

If the StackSets deployment to an account fails and the message is "IAM role exists", delete the IAM role from the member account and then retry the role deployment in the management account.

**To retry the IAM role deployments**

1. Sign in to the management account.
2. Open the Network Manager console at https://console.aws.amazon.com/networkmanager/home.
3. From the navigation pane, choose **Reachability Analyzer**, **Settings**.
4. Under **IAM role deployments status**, choose **Retry role deployment**. The deployments can take several minutes to complete, depending on the number of member accounts in your organization.

For a message other than "IAM role exists", open a case with AWS Support. For more information, see Create a support case in the *AWS Support Guide*.

# Quotas for Reachability 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 Reachability 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 Reachability Analyzer.

| Name | Default | Adjustable |
| --- | --- | --- |
| Paths | 1,000 | Yes |
| Analyses | 10,000 | Yes |
| Concurrent analyses | 100 | Yes |

# Document history for Reachability Analyzer

The following table describes the releases for Reachability Analyzer.

| Change | Description | Date |
| --- | --- | --- |
| New feature (p. 47) | You can specify VPC endpoints as sources and destinations, and Network Firewall firewalls as intermediate path components. | March 21, 2023 |
| Multi-account support (p. 47) | Reachability Analyzer supports reachability analysis between AWS resources in different AWS accounts within an organization from AWS Organizations. | November 27, 2022 |
| New feature (p. 47) | You can specify transit gateways as sources, destinations, and intermediate path components. | March 25, 2022 |
| Initial release (p. 47) | This release introduces Reachability Analyzer. | December 10, 2020 |