



AWS PrivateLink

Amazon Virtual Private Cloud



Amazon Virtual Private Cloud: AWS PrivateLink

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

What is AWS PrivateLink?	1
Use cases	1
Work with VPC endpoints	2
Pricing	3
Concepts	3
Architecture diagram	3
Service providers	4
Service consumers	5
AWS PrivateLink connections	7
Private hosted zones	7
Get started	8
Step 1: Create a VPC with subnets	9
Step 2: Launch the instances	9
Step 3: Test CloudWatch access	11
Step 4: Create a VPC endpoint to access CloudWatch	12
Step 5: Test the VPC endpoint	12
Step 6: Clean up	13
Access AWS services	14
Overview	15
DNS hostnames	16
DNS resolution	18
Private DNS	18
Subnets and Availability Zones	18
IP address types	22
Services that integrate	23
View available AWS service names	37
View information about a service	37
View endpoint policy support	39
View IPv6 support	41
Create an interface endpoint	42
Prerequisites	42
Create a VPC endpoint	43
Shared subnets	44
Configure an interface endpoint	44

Add or remove subnets	45
Associate security groups	46
Edit the VPC endpoint policy	46
Enable private DNS names	47
Manage tags	48
Receive alerts for interface endpoint events	48
Create an SNS notification	48
Add an access policy	49
Add a key policy	50
Delete an interface endpoint	51
Gateway endpoints	51
Overview	52
Routing	53
Security	54
Endpoints for Amazon S3	55
Endpoints for DynamoDB	65
Access SaaS products	73
Overview	73
Create an interface endpoint	74
Access virtual appliances	76
Overview	76
IP address types	78
Routing	79
Create a Gateway Load Balancer endpoint service	80
Considerations	80
Prerequisites	81
Create the endpoint service	81
Make your endpoint service available	82
Create a Gateway Load Balancer endpoint	82
Considerations	83
Prerequisites	84
Create the endpoint	84
Configure routing	85
Manage tags	86
Delete the endpoint	87
Share your services	88

Overview	88
DNS hostnames	89
Private DNS	90
IP address types	90
Create an endpoint service	91
Considerations	92
Prerequisites	92
Create an endpoint service	93
Make your endpoint service available to service consumers	94
Configure an endpoint service	96
Manage permissions	96
Accept or reject connection requests	97
Change the load balancer association	99
Associate a private DNS name	99
Modify the supported IP address types	101
Manage tags	101
Manage DNS names	103
Domain ownership verification	103
Get the name and value	104
Add a TXT record to your domain's DNS server	105
Check whether the TXT record is published	106
Troubleshoot domain verification issues	107
Receive alerts for endpoint service events	108
Create an SNS notification	108
Add an access policy	109
Add a key policy	110
Delete an endpoint service	110
Identity and access management	112
Audience	112
Authenticating with identities	113
AWS account root user	113
Federated identity	114
IAM users and groups	114
IAM roles	114
Managing access using policies	116
Identity-based policies	117

Resource-based policies	117
Access control lists (ACLs)	117
Other policy types	118
Multiple policy types	118
How AWS PrivateLink works with IAM	118
Identity-based policies	119
Resource-based policies	120
Policy actions	121
Policy resources	122
Policy condition keys	122
ACLs	123
ABAC	123
Temporary credentials	124
Principal permissions	124
Service roles	125
Service-linked roles	125
Identity-based policy examples	125
Control the use of VPC endpoints	126
Control VPC endpoints creation based on the service owner	126
Control the private DNS names that can be specified for VPC endpoint services	127
Control the service names that can be specified for VPC endpoint services	128
Endpoint policies	129
Considerations	129
Default endpoint policy	130
Policies for interface endpoints	130
Principals for gateway endpoints	130
Update a VPC endpoint policy	131
CloudWatch metrics	132
Endpoint metrics and dimensions	132
Endpoint service metrics and dimensions	135
View the CloudWatch metrics	138
Use built-in Contributor Insights rules	139
Enable Contributor Insights rules	140
Disable Contributor Insights rules	141
Delete Contributor Insights rules	142
Quotas	143

Document history 145

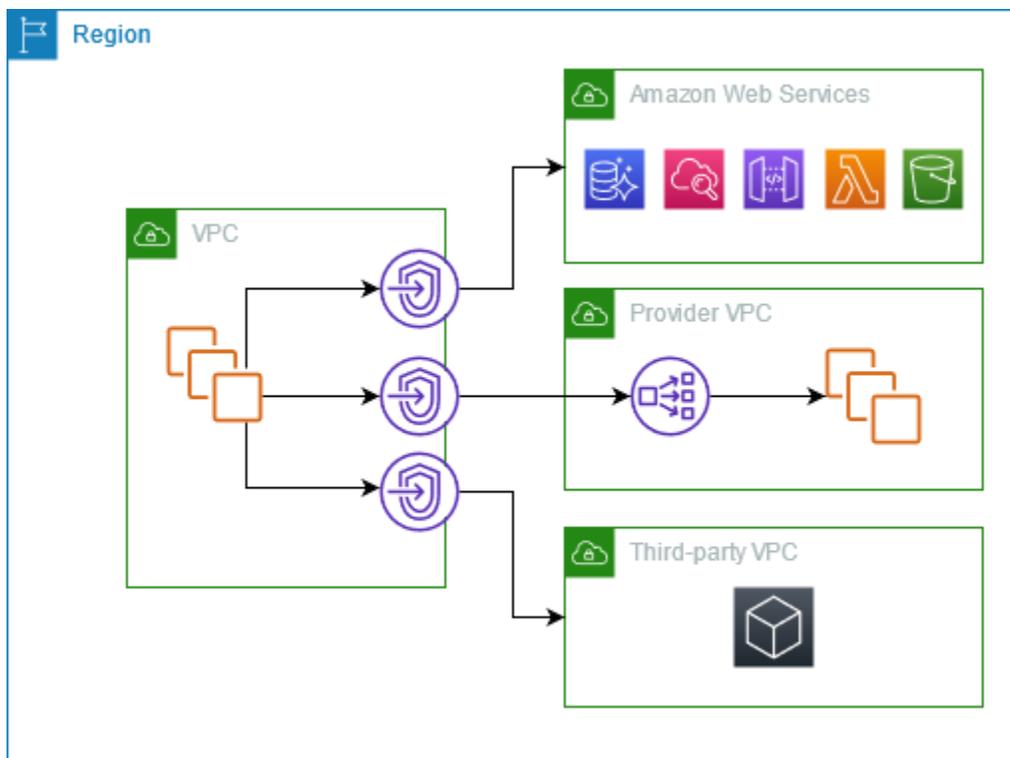
What is AWS PrivateLink?

AWS PrivateLink is a highly available, scalable technology that you can use to privately connect your VPC to services as if they were in your VPC. You do not need to use an internet gateway, NAT device, public IP address, AWS Direct Connect connection, or AWS Site-to-Site VPN connection to allow communication with the service from your private subnets. Therefore, you control the specific API endpoints, sites, and services that are reachable from your VPC.

Use cases

You can create VPC endpoints to connect resources in your VPC to services that integrate with AWS PrivateLink. You can create your own VPC endpoint service and make it available to other AWS customers. For more information, see [the section called "Concepts"](#).

In the following diagram, the VPC on the left has several EC2 instances in a private subnet and three interface VPC endpoints. The top-most VPC endpoint connects to an AWS service. The middle VPC endpoint connects to a service hosted by another AWS account (a VPC endpoint service). The bottom VPC endpoint connects to an AWS Marketplace partner service.



Learn more

- [the section called “Concepts”](#)
- [Access AWS services](#)
- [Access SaaS products](#)
- [Access virtual appliances](#)
- [Share your services](#)

Work with VPC endpoints

You can create, access, and manage VPC endpoints using any of the following:

- **AWS Management Console** — Provides a web interface that you can use to access your AWS PrivateLink resources.
- **AWS Command Line Interface (AWS CLI)** — Provides commands for a broad set of AWS services, including AWS PrivateLink. For more information about commands for AWS PrivateLink, see [ec2](#) in the *AWS CLI Command Reference*.
- **AWS CloudFormation** - Create templates that describe your AWS resources. You use the templates to provision and manage these resources as a single unit. For more information, see the following AWS PrivateLink resources:
 - [AWS::EC2::VPCEndpoint](#)
 - [AWS::EC2::VPCEndpointConnectionNotification](#)
 - [AWS::EC2::VPCEndpointService](#)
 - [AWS::EC2::VPCEndpointServicePermissions](#)
 - [AWS::ElasticLoadBalancingV2::LoadBalancer](#)
- **AWS SDKs** — Provide language-specific APIs. The SDKs take 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 Amazon VPC. However, it requires that your application handle low-level details such as generating the hash to sign the request and handling errors. For more information, see [AWS PrivateLink actions](#) in the *Amazon EC2 API Reference*.

Pricing

For information about the pricing for VPC endpoints, see [AWS PrivateLink Pricing](#).

AWS PrivateLink concepts

You can use Amazon VPC to define a virtual private cloud (VPC), which is a logically isolated virtual network. You can launch AWS resources in your VPC. You can allow the resources in your VPC to connect to resources outside that VPC. For example, add an internet gateway to the VPC to allow access to the internet, or add a VPN connection to allow access to your on-premises network. Alternatively, use AWS PrivateLink to allow the resources in your VPC to connect to services in other VPCs using private IP addresses, as if those services were hosted directly in your VPC.

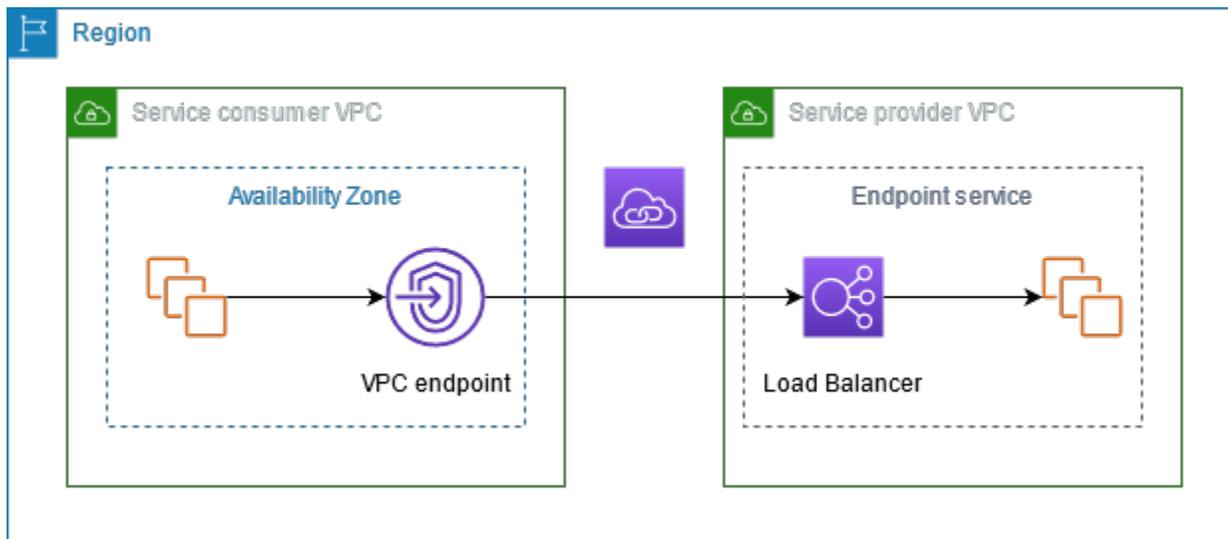
The following are important concepts to understand as you get started using AWS PrivateLink.

Contents

- [Architecture diagram](#)
- [Service providers](#)
- [Service consumers](#)
- [AWS PrivateLink connections](#)
- [Private hosted zones](#)

Architecture diagram

The following diagram provides a high-level overview of how AWS PrivateLink works. Service consumers create interface VPC endpoints to connect to endpoint services that are hosted by service providers.



Service providers

The owner of a service is the *service provider*. Service providers include AWS, AWS Partners, and other AWS accounts. Service providers can host their services using AWS resources, such as EC2 instances, or using on-premises servers.

Concepts

- [Endpoint services](#)
- [Service names](#)
- [Service states](#)

Endpoint services

A service provider creates an *endpoint service* to make their service available in a Region. A service provider must specify a load balancer when creating an endpoint service. The load balancer receives requests from service consumers and routes them to your service.

By default, your endpoint service is not available to service consumers. You must add permissions that allow specific AWS principals to connect to your endpoint service.

Service names

Each endpoint service is identified by a service name. A service consumer must specify the name of the service when creating a VPC endpoint. Service consumers can query the service names for AWS services. Service providers must share the names of their services with service consumers.

Service states

The following are the possible states for an endpoint service:

- `Pending` - The endpoint service is being created.
- `Available` - The endpoint service is available.
- `Failed` - The endpoint service could not be created.
- `Deleting` - The service provider deleted the endpoint service and deletion is in progress.
- `Deleted` - The endpoint service is deleted.

Service consumers

The user of a service is a *service consumer*. Service consumers can access endpoint services from AWS resources, such as EC2 instances, or from on-premises servers.

Concepts

- [VPC endpoints](#)
- [Endpoint network interfaces](#)
- [Endpoint policies](#)
- [Endpoint states](#)

VPC endpoints

A service consumer creates a *VPC endpoint* to connect their VPC to an endpoint service. A service consumer must specify the service name of the endpoint service when creating a VPC endpoint. There are multiple types of VPC endpoints. You must create the type of VPC endpoint that's required by the endpoint service.

- `Interface` - Create an *interface endpoint* to send TCP traffic to an endpoint service. Traffic destined for the endpoint service is resolved using DNS.
- `GatewayLoadBalancer` - Create a *Gateway Load Balancer endpoint* to send traffic to a fleet of virtual appliances using private IP addresses. You route traffic from your VPC to the Gateway Load Balancer endpoint using route tables. The Gateway Load Balancer distributes traffic to the virtual appliances and can scale with demand.

There is another type of VPC endpoint, Gateway, which creates a *gateway endpoint* to send traffic to Amazon S3 or DynamoDB. Gateway endpoints do not use AWS PrivateLink, unlike the other types of VPC endpoints. For more information, see [the section called "Gateway endpoints"](#).

Endpoint network interfaces

An *endpoint network interface* is a requester-managed network interface that serves as an entry point for traffic destined to an endpoint service. For each subnet that you specify when you create a VPC endpoint, we create an endpoint network interface in the subnet.

If a VPC endpoint supports IPv4, its endpoint network interfaces have IPv4 addresses. If a VPC endpoint supports IPv6, its endpoint network interfaces have IPv6 addresses. The IPv6 address for an endpoint network interface is unreachable from the internet. When you describe an endpoint network interface with an IPv6 address, notice that `denyAllIgwTraffic` is enabled.

The IP addresses of an endpoint network interface will not change during the lifetime of its VPC endpoint.

Endpoint policies

A *VPC endpoint policy* is an IAM resource policy that you attach to a VPC endpoint. It determines which principals can use the VPC endpoint to access the endpoint service. The default VPC endpoint policy allows all actions by all principals on all resources over the VPC endpoint.

Endpoint states

When you create a VPC endpoint, the endpoint service receives a connection request. The service provider can accept or reject the request. If the service provider accepts the request, the service consumer can use the VPC endpoint after it enters the `Available` state.

The following are the possible states for a VPC endpoint:

- `PendingAcceptance` - The connection request is pending. This is the initial state if requests are manually accepted.
- `Pending` - The service provider accepted the connection request. This is the initial state if requests are automatically accepted. The VPC endpoint returns to this state if the service consumer modifies the VPC endpoint.
- `Available` - The VPC endpoint is available for use.
- `Rejected` - The service provider rejected the connection request. The service provider can also reject a connection after it is available for use.

- **Expired** - The connection request expired.
- **Failed** - The VPC endpoint could not be made available.
- **Deleting** - The service consumer deleted the VPC endpoint and deletion is in progress.
- **Deleted** - The VPC endpoint is deleted.

AWS PrivateLink connections

Traffic from your VPC is sent to an endpoint service using a connection between the VPC endpoint and the endpoint service. Traffic between a VPC endpoint and an endpoint service stays within the AWS network, without traversing the public internet.

A service provider adds [permissions](#) so that service consumers can access the endpoint service. The service consumer initiates the connection and the service provider accepts or rejects the connection request.

With interface VPC endpoints, service consumers can use [endpoint policies](#) to control which IAM principals can use a VPC endpoint to access an endpoint service.

Private hosted zones

A *hosted zone* is a container for DNS records that define how to route traffic for a domain or subdomain. With a *public hosted zone*, the records specify how to route traffic on the internet. With a *private hosted zone*, the records specify how to route traffic in your VPCs.

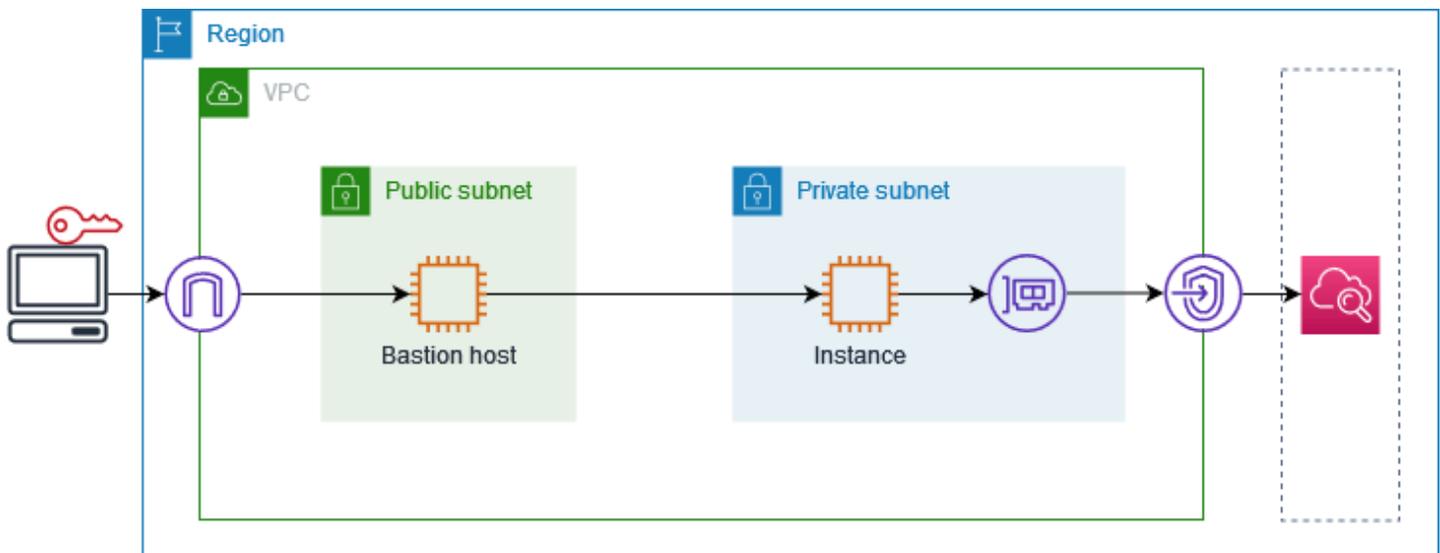
You can configure Amazon Route 53 to route domain traffic to a VPC endpoint. For more information, see [Routing traffic to a VPC endpoint using your domain name](#).

You can use Route 53 to configure split-horizon DNS, where you use the same domain name for both a public website and an endpoint service powered by AWS PrivateLink. DNS requests for the public hostname from the consumer VPC resolve to the private IP addresses of the endpoint network interfaces, but requests from outside the VPC continue to resolve to the public endpoints. For more information, see [DNS Mechanisms for Routing Traffic and Enabling Failover for AWS PrivateLink Deployments](#).

Get started with AWS PrivateLink

This tutorial demonstrates how to send a request from an EC2 instance in a private subnet to Amazon CloudWatch using AWS PrivateLink.

The following diagram provides an overview of this scenario. To connect from your computer to the instance in the private subnet, you'll first connect to a bastion host in a public subnet. Both the bastion host and the instance must use the same key pair. Because the `.pem` file for the private key is on your computer, not the bastion host, you'll use SSH key forwarding. Then, you can connect to the instance from the bastion host without specifying the `.pem` file in the `ssh` command. After you set up a VPC endpoint for CloudWatch, traffic from the instance that's destined for CloudWatch is resolved to the endpoint network interface and then sent to CloudWatch using the VPC endpoint.



For testing purposes, you can use a single Availability Zone. In production, we recommend that you use at least two Availability Zones for low latency and high availability.

Tasks

- [Step 1: Create a VPC with subnets](#)
- [Step 2: Launch the instances](#)
- [Step 3: Test CloudWatch access](#)
- [Step 4: Create a VPC endpoint to access CloudWatch](#)
- [Step 5: Test the VPC endpoint](#)
- [Step 6: Clean up](#)

Step 1: Create a VPC with subnets

Use the following procedure to create a VPC with a public subnet and a private subnet.

To create the VPC

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. Choose **Create VPC**.
3. For **Resources to create**, choose **VPC and more**.
4. For **Name tag auto-generation**, enter a name for the VPC.
5. To configure the subnets, do the following:
 - a. For **Number of Availability Zones**, choose **1** or **2**, depending on your needs.
 - b. For **Number of public subnets**, ensure that you have one public subnet per Availability Zone.
 - c. For **Number of private subnets**, ensure that you have one private subnet per Availability Zone.
6. Choose **Create VPC**.

Step 2: Launch the instances

Using the VPC that you created in the previous step, launch the bastion host in the public subnet and the instance in the private subnet.

Prerequisites

- Create a key pair using the **.pem** format. You must choose this key pair when you launch both the bastion host and the instance.
- Create a security group for the bastion host that allows inbound SSH traffic from the CIDR block for your computer.
- Create a security group for the instance that allows inbound SSH traffic from the security group for the bastion host.
- Create an IAM instance profile and attach the **CloudWatchReadOnlyAccess** policy.

To launch the bastion host

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch instance**.
3. For **Name**, enter a name for your bastion host.
4. Keep the default image and instance type.
5. For **Key pair**, select your key pair.
6. For **Network settings**, do the following:
 - a. For **VPC**, choose your VPC.
 - b. For **Subnet**, choose the public subnet.
 - c. For **Auto-assign public IP**, choose **Enable**.
 - d. For **Firewall**, choose **Select existing security group** and then choose the security group for the bastion host.
7. Choose **Launch instance**.

To launch the instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch instance**.
3. For **Name**, enter a name for your instance.
4. Keep the default image and instance type.
5. For **Key pair**, select your key pair.
6. For **Network settings**, do the following:
 - a. For **VPC**, choose your VPC.
 - b. For **Subnet**, choose the private subnet.
 - c. For **Auto-assign public IP**, choose **Disable**.
 - d. For **Firewall**, choose **Select existing security group** and then choose the security group for the instance.
7. Expand **Advanced details**. For **IAM instance profile**, choose your IAM instance profile.
8. Choose **Launch instance**.

Step 3: Test CloudWatch access

Use the following procedure to confirm that the instance can't access CloudWatch. You'll do so using a read-only AWS CLI command for CloudWatch.

To test CloudWatch access

1. From your computer, add the key pair to the SSH agent using the following command, where *key.pem* is the name of your .pem file.

```
ssh-add ./key.pem
```

If you receive an error that permissions for your key pair are too open, run the following command, and then retry the previous command.

```
chmod 400 ./key.pem
```

2. Connect to the bastion host from your computer. You must specify the `-A` option, the instance user name (for example, `ec2-user`), and the public IP address of the bastion host.

```
ssh -A ec2-user@bastion-public-ip-address
```

3. Connect to the instance from the bastion host. You must specify the instance user name (for example, `ec2-user`) and the private IP address of the instance.

```
ssh ec2-user@instance-private-ip-address
```

4. Run the CloudWatch [list-metrics](#) command on the instance as follows. For the `--region` option, specify the Region where you created the VPC.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --region us-east-1
```

5. After a few minutes, the command times out. This demonstrates that you can't access CloudWatch from the instance with the current VPC configuration.

```
Connect timeout on endpoint URL: https://monitoring.us-east-1.amazonaws.com/
```

6. Stay connected to your instance. After you create the VPC endpoint, you'll try this `list-metrics` command again.

Step 4: Create a VPC endpoint to access CloudWatch

Use the following procedure to create a VPC endpoint that connects to CloudWatch.

Prerequisite

Create a security group for the VPC endpoint that allows traffic to CloudWatch. For example, add a rule that allows HTTPS traffic from the VPC CIDR block.

To create a VPC endpoint for CloudWatch

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create endpoint**.
4. For **Name tag**, enter a name for the endpoint.
5. For **Service category**, choose **AWS services**.
6. For **Service**, select **com.amazonaws.*region*.monitoring**.
7. For **VPC**, select your VPC.
8. For **Subnets**, select the Availability Zone and then select the private subnet.
9. For **Security group**, select the security group for the VPC endpoint.
10. For **Policy**, select **Full access** to allow all operations by all principals on all resources over the VPC endpoint.
11. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
12. Choose **Create endpoint**. The initial status is **Pending**. Before you go to the next step, wait until the status is **Available**. This can take a few minutes.

Step 5: Test the VPC endpoint

Verify that the VPC endpoint is sending requests from your instance to CloudWatch.

To test the VPC endpoint

Run the following command on your instance. For the `--region` option, specify the Region where you created the VPC endpoint.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --region us-east-1
```

If you get a response, even a response with empty results, then you are connected to CloudWatch using AWS PrivateLink.

If you get an `UnauthorizedOperation` error, ensure that the instance has an IAM role that allows access to CloudWatch.

If the request times out, verify the following:

- The security group for the endpoint allows traffic to CloudWatch.
- The `--region` option specifies the Region in which you created the VPC endpoint.

Step 6: Clean up

If you no longer need the bastion host and instance that you created for this tutorial, you can terminate them.

To terminate the instances

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select both test instances and choose **Instance state, Terminate instance**.
4. When prompted for confirmation, choose **Terminate**.

If you no longer need the VPC endpoint, you can delete it.

To delete the VPC endpoint

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the VPC endpoint.
4. Choose **Actions, Delete VPC endpoints**.
5. When prompted for confirmation, enter **delete** and then choose **Delete**.

Access AWS services through AWS PrivateLink

You access an AWS service using an endpoint. The default service endpoints are public interfaces, so you must add an internet gateway to your VPC so that traffic can get from the VPC to the AWS service. If this configuration doesn't work with your network security requirements, you can use AWS PrivateLink to connect your VPC to AWS services as if they were in your VPC, without the use of an internet gateway.

You can privately access the AWS services that integrate with AWS PrivateLink using VPC endpoints. You can build and manage all layers of your application stack without using an internet gateway.

Pricing

You are billed for each hour that your interface VPC endpoint is provisioned in each Availability Zone. You are also billed per GB of data processed. For more information, see [AWS PrivateLink Pricing](#).

Contents

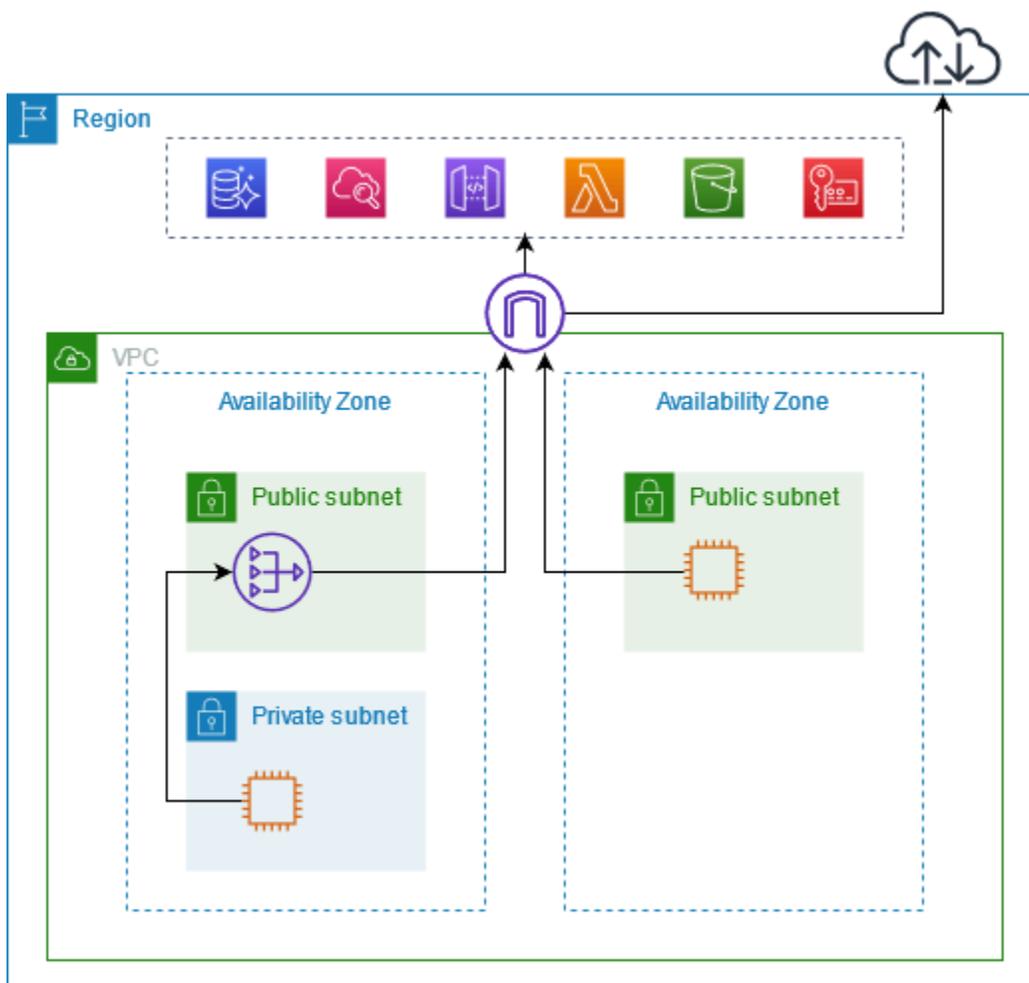
- [Overview](#)
- [DNS hostnames](#)
- [DNS resolution](#)
- [Private DNS](#)
- [Subnets and Availability Zones](#)
- [IP address types](#)
- [AWS services that integrate with AWS PrivateLink](#)
- [Access an AWS service using an interface VPC endpoint](#)
- [Configure an interface endpoint](#)
- [Receive alerts for interface endpoint events](#)
- [Delete an interface endpoint](#)
- [Gateway endpoints](#)

Overview

You can access AWS services through their public service endpoints or connect to supported AWS services using AWS PrivateLink. This overview compares these methods.

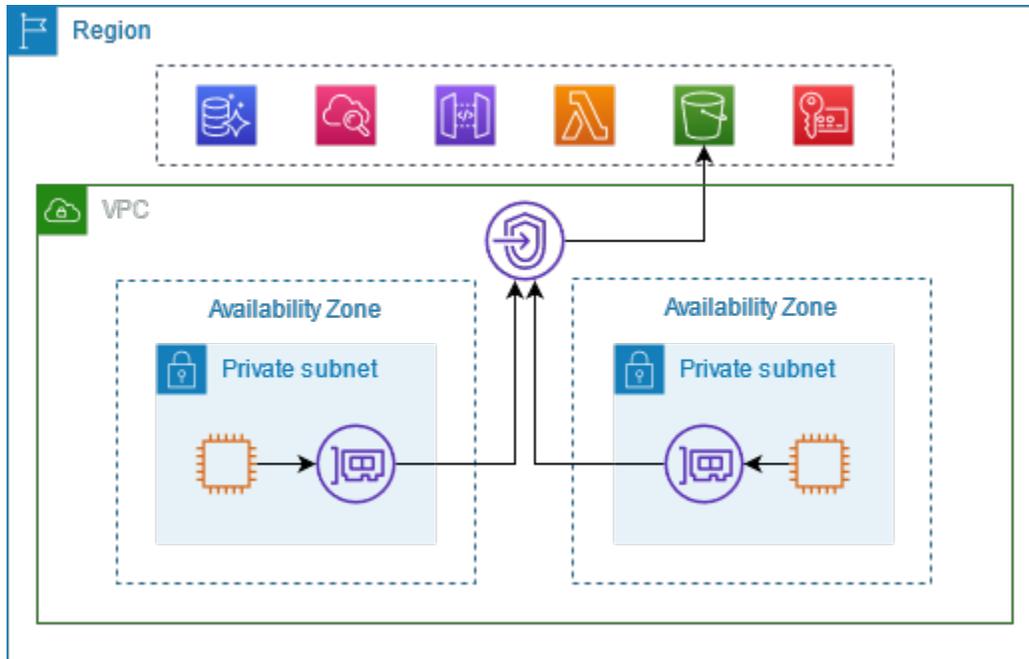
Access through public service endpoints

The following diagram shows how instances access AWS services through the public service endpoints. Traffic to an AWS service from an instance in a public subnet is routed to the internet gateway for the VPC and then to the AWS service. Traffic to an AWS service from an instance in a private subnet is routed to a NAT gateway, then to the internet gateway for the VPC, and then to the AWS service. While this traffic traverses the internet gateway, it does not leave the AWS network.



Connect through AWS PrivateLink

The following diagram shows how instances access AWS services through AWS PrivateLink. First, you create an interface VPC endpoint, which establishes connections between the subnets in your VPC and an AWS service using network interfaces. Traffic destined for the AWS service is resolved to the private IP addresses of the endpoint network interfaces using DNS, and then sent to the AWS service using the connection between the VPC endpoint and the AWS service.



AWS services accept connection requests automatically. The service can't initiate requests to resources through the VPC endpoint.

DNS hostnames

Most AWS services offer public Regional endpoints, which have the following syntax.

```
protocol://service_code.region_code.amazonaws.com
```

For example, the public endpoint for Amazon CloudWatch in us-east-2 is as follows.

```
https://monitoring.us-east-2.amazonaws.com
```

With AWS PrivateLink, you send traffic to the service using private endpoints. When you create an interface VPC endpoint, we create Regional and zonal DNS names that you can use to communicate with the AWS service from your VPC.

The Regional DNS name for your interface VPC endpoint has the following syntax:

```
endpoint_id.service_id.region.vpce.amazonaws.com
```

The zonal DNS names have the following syntax:

```
endpoint_id-az_name.service_id.region.vpce.amazonaws.com
```

When you create an interface VPC endpoint for an AWS service, you can enable [private DNS](#). With private DNS, you can continue to make requests to a service using the DNS name for its public endpoint, while leveraging private connectivity through the interface VPC endpoint. For more information, see [the section called "DNS resolution"](#).

The following [describe-vpc-endpoints](#) command displays the DNS entries for an interface endpoint.

```
aws ec2 describe-vpc-endpoints --vpc-endpoint-id vpce-099deb00b40f00e22 --query  
VpcEndpoints[*].DnsEntries
```

The following is example output for an interface endpoint for Amazon CloudWatch with private DNS names enabled. The first entry is the private Regional endpoint. The next three entries are the private zonal endpoints. The final entry is from the hidden private hosted zone, which resolves requests to the public endpoint to the private IP addresses of the endpoint network interfaces.

```
[  
  [  
    {  
      "DnsName": "vpce-099deb00b40f00e22-lj2wisx3.monitoring.us-  
east-2.vpce.amazonaws.com",  
      "HostedZoneId": "ZC8PG0KIFKBRI"  
    },  
    {  
      "DnsName": "vpce-099deb00b40f00e22-lj2wisx3-us-east-2c.monitoring.us-  
east-2.vpce.amazonaws.com",  
      "HostedZoneId": "ZC8PG0KIFKBRI"  
    },  
    {  
      "DnsName": "vpce-099deb00b40f00e22-lj2wisx3-us-east-2a.monitoring.us-  
east-2.vpce.amazonaws.com",  
      "HostedZoneId": "ZC8PG0KIFKBRI"  
    },  
    {
```

```
        "DnsName": "vpce-099deb00b40f00e22-1j2wisx3-us-east-2b.monitoring.us-  
east-2.vpce.amazonaws.com",  
        "HostedZoneId": "ZC8PG0KIFKBRI"  
    },  
    {  
        "DnsName": "monitoring.us-east-2.amazonaws.com",  
        "HostedZoneId": "Z06320943MM0WYG6MAVL9"  
    }  
]  
]
```

DNS resolution

The DNS records that we create for your interface VPC endpoint are public. Therefore, these DNS names are publicly resolvable. However, DNS requests from outside the VPC still return the private IP addresses of the endpoint network interfaces, so these IP addresses can't be used to access the endpoint service unless you have access to the VPC.

Private DNS

If you enable private DNS for your interface VPC endpoint, and your VPC has both [DNS hostnames and DNS resolution](#) enabled, we create a hidden, AWS-managed private hosted zone for you. The hosted zone contains a record set for the default DNS name for the service that resolves it to the private IP addresses of the endpoint network interfaces in your VPC. Therefore, if you have existing applications that send requests to the AWS service using a public Regional endpoint, those requests now go through the endpoint network interfaces, without requiring that you make any changes to those applications.

Amazon provides a DNS server for your VPC, called the [Route 53 Resolver](#). The Route 53 Resolver automatically resolves local VPC domain names and record in private hosted zones. However, you can't use the Route 53 Resolver from outside your VPC. If you'd like to access your VPC endpoint from your on-premises network, you can use Route 53 Resolver endpoints and Resolver rules. For more information, see [Integrating AWS Transit Gateway with AWS PrivateLink and Amazon Route 53 Resolver](#).

Subnets and Availability Zones

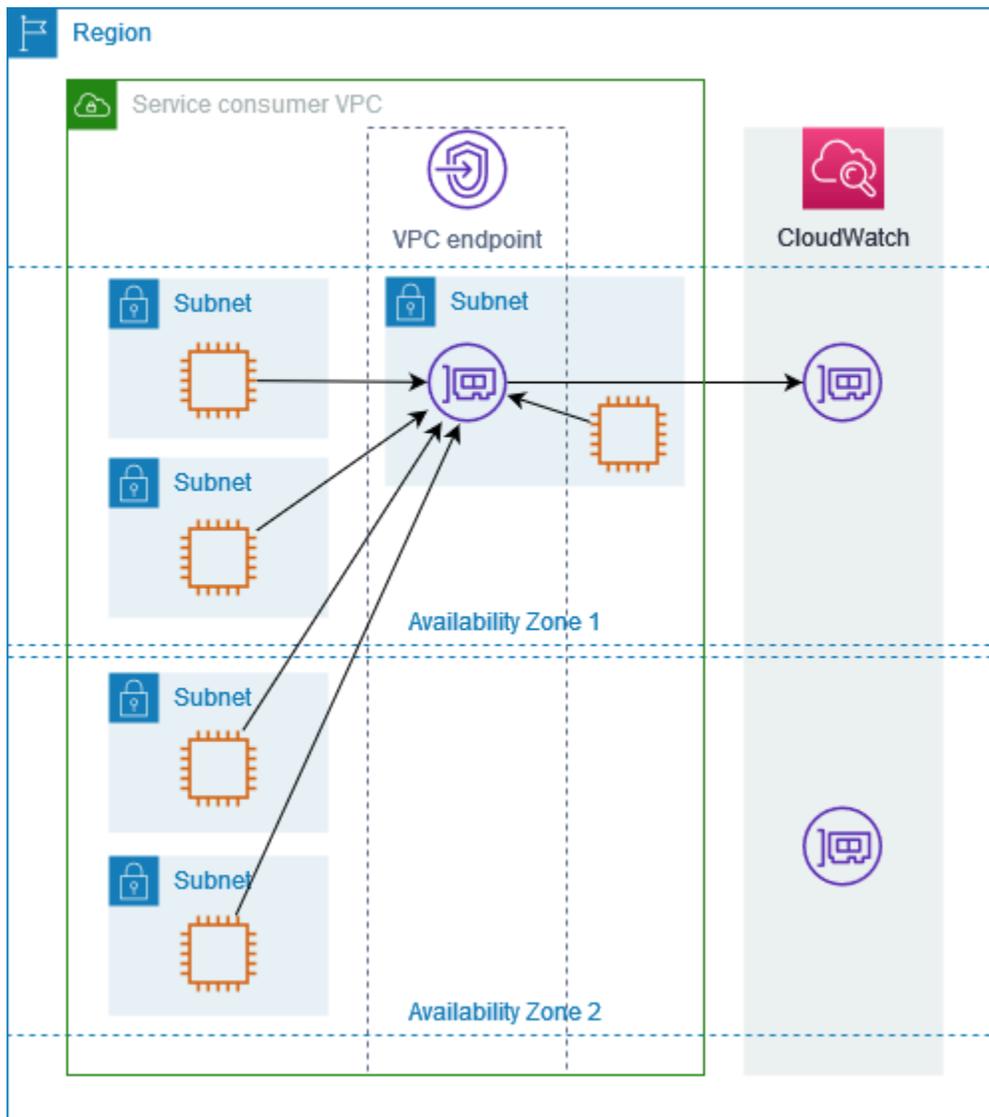
You can configure your VPC endpoint with one subnet per Availability Zone. We create an endpoint network interface for the VPC endpoint in your subnet. We assign IP addresses to each endpoint

network interface from its subnet, based on the [IP address type](#) of the VPC endpoint. The IP addresses of an endpoint network interface will not change during the lifetime of its VPC endpoint.

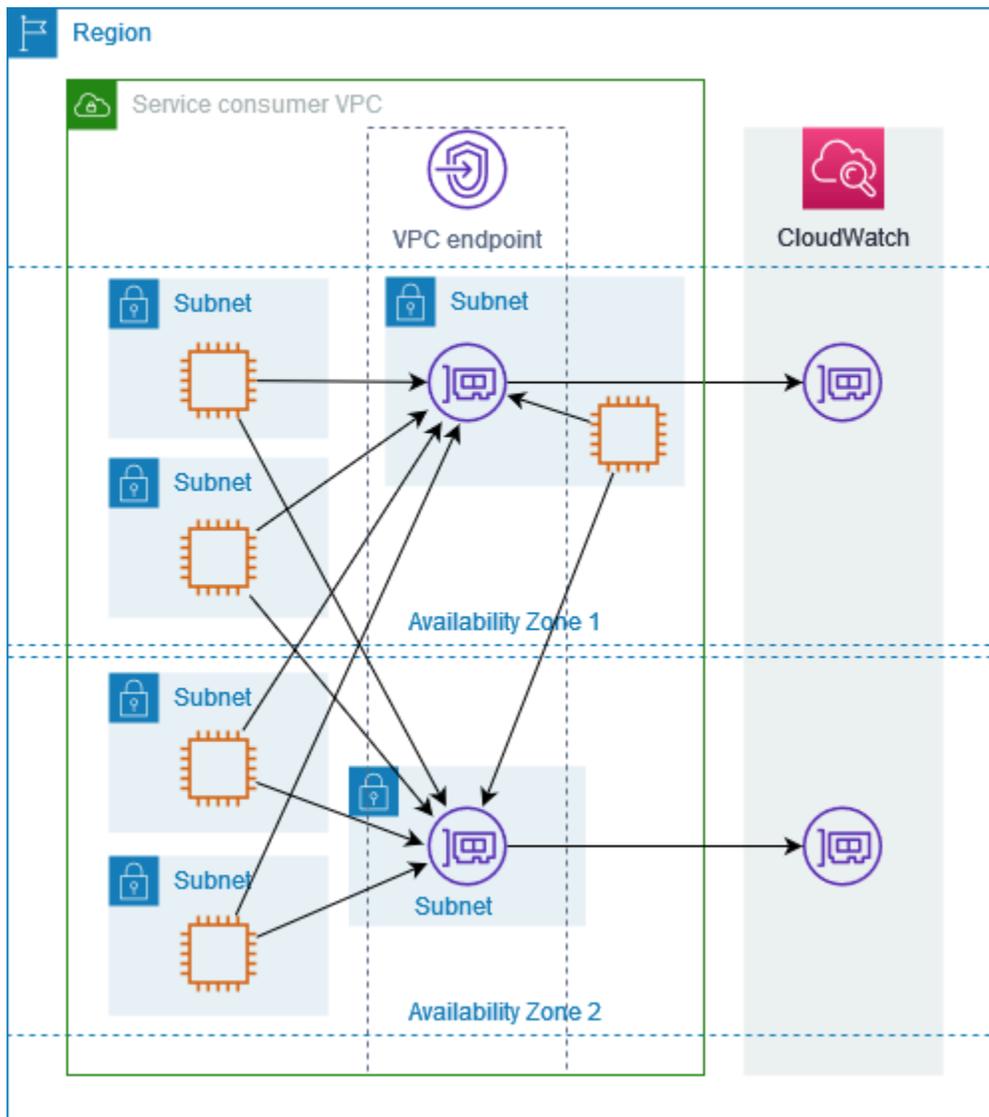
In a production environment, for high availability and resiliency, we recommend the following:

- Configure at least two Availability Zones per VPC endpoint and deploy your AWS resources that must access the AWS service in these Availability Zones.
- Configure private DNS names for the VPC endpoint.
- Access the AWS service by using its Regional DNS name, also known as the public endpoint.

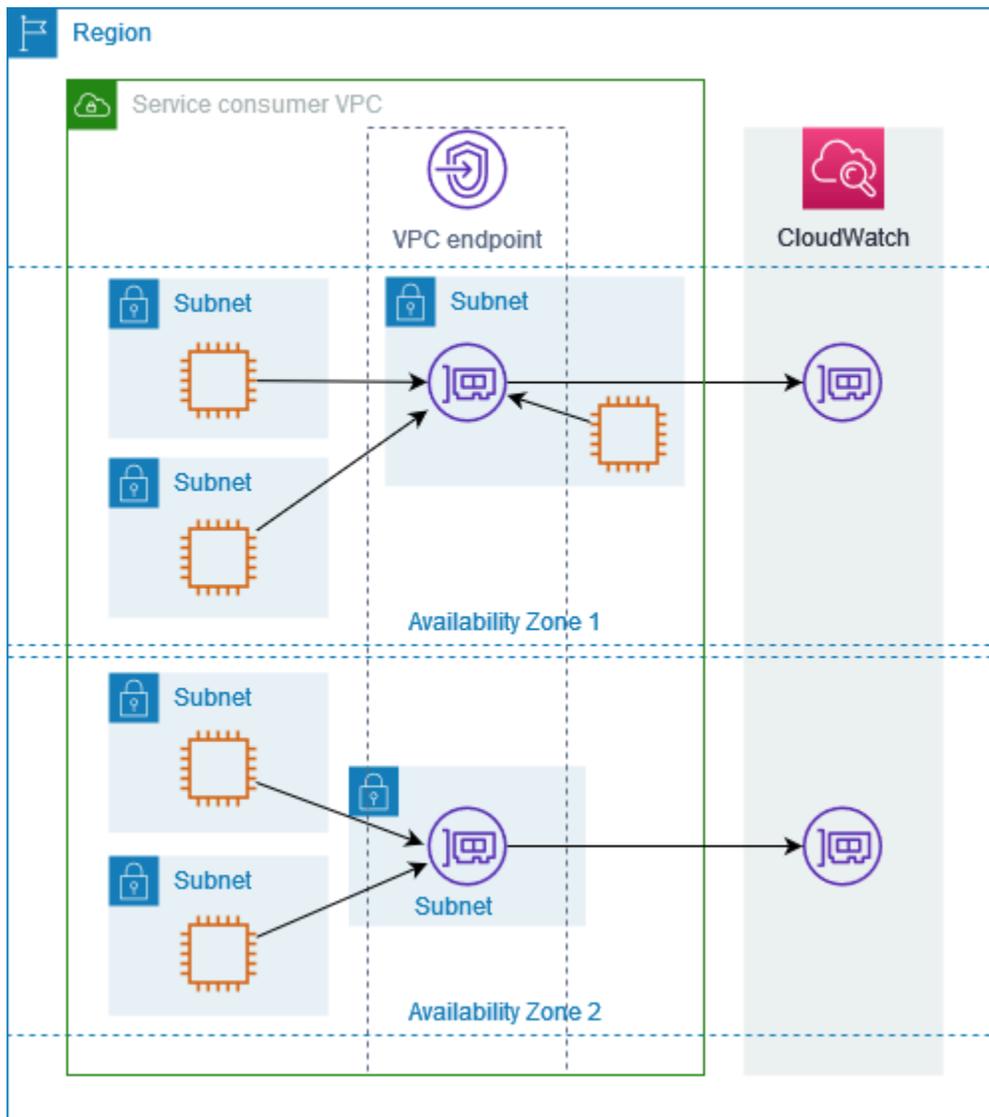
The following diagram shows a VPC endpoint for Amazon CloudWatch with an endpoint network interface in a single Availability Zone. When any resource in any subnet in the VPC accesses Amazon CloudWatch using its public endpoint, we resolve the traffic to the IP address of the endpoint network interface. This includes traffic from subnets in other Availability Zones. However, if Availability Zone 1 is impaired, the resources in Availability Zone 2 lose access to Amazon CloudWatch.



The following diagram shows a VPC endpoint for Amazon CloudWatch with endpoint network interfaces in two Availability Zones. When any resource in any subnet in the VPC accesses Amazon CloudWatch by using its public endpoint, we select a healthy endpoint network interface, using the round robin algorithm to alternate between them. We then resolve the traffic to the IP address of the selected endpoint network interface.



If it's better for your use case, you can send traffic from your resources to the AWS service by using the endpoint network interface in the same Availability Zone. To do so, use the private zonal endpoint or IP address of the endpoint network interface.



IP address types

AWS services can support IPv6 through their private endpoints even if they do not support IPv6 through their public endpoints. Endpoints that support IPv6 can respond to DNS queries with AAAA records.

Requirements to enable IPv6 for an interface endpoint

- The AWS service must make its service endpoints available over IPv6. For more information, see [the section called “View IPv6 support”](#).
- The IP address type of an interface endpoint must be compatible with the subnets for the interface endpoint, as described here:

- **IPv4** – Assign IPv4 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have IPv4 address ranges.
- **IPv6** – Assign IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets are IPv6 only subnets.
- **Dualstack** – Assign both IPv4 and IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have both IPv4 and IPv6 address ranges.

If an interface VPC endpoint supports IPv4, the endpoint network interfaces have IPv4 addresses. If an interface VPC endpoint supports IPv6, the endpoint network interfaces have IPv6 addresses. The IPv6 address for an endpoint network interface is unreachable from the internet. If you describe an endpoint network interface with an IPv6 address, notice that `denyAllIgwTraffic` is enabled.

AWS services that integrate with AWS PrivateLink

The following AWS services integrate with AWS PrivateLink. You can create a VPC endpoint to connect to these services privately, as if they were running in your own VPC.

Choose the link in the **AWS service** column to see the documentation for services that integrate with AWS PrivateLink. The **Service name** column contains the service name that you specify when you create the interface VPC endpoint, or indicates that the service manages the endpoint.

AWS service	Service name
Access Analyzer	com.amazonaws. <i>region</i> .access-analyzer
AWS Account Management	com.amazonaws. <i>region</i> .account
Amazon API Gateway	com.amazonaws. <i>region</i> .execute-api
AWS AppConfig	com.amazonaws. <i>region</i> .appconfig com.amazonaws. <i>region</i> .appconfigdata
AWS App Mesh	com.amazonaws. <i>region</i> .appmesh com.amazonaws. <i>region</i> .appmesh-envoy-management

AWS service	Service name
AWS App Runner	com.amazonaws. <i>region</i> .apprunner
AWS App Runner services	com.amazonaws. <i>region</i> .apprunner.requests
Application Auto Scaling	com.amazonaws. <i>region</i> .application-autoscaling
AWS Application Migration Service	com.amazonaws. <i>region</i> .mgn
Amazon AppStream 2.0	com.amazonaws. <i>region</i> .appstream.api
	com.amazonaws. <i>region</i> .appstream.streaming
AWS AppSync	com.amazonaws. <i>region</i> .appsync-api
Amazon Athena	com.amazonaws. <i>region</i> .athena
AWS Audit Manager	com.amazonaws. <i>region</i> .auditmanager
Amazon Aurora	com.amazonaws. <i>region</i> .rds
AWS Auto Scaling	com.amazonaws. <i>region</i> .autoscaling-plans
AWS B2B Data Interchange	com.amazonaws. <i>region</i> .b2bi
AWS Backup	com.amazonaws. <i>region</i> .backup
	com.amazonaws. <i>region</i> .backup-gateway
AWS Batch	com.amazonaws. <i>region</i> .batch
Amazon Bedrock	com.amazonaws. <i>region</i> .bedrock
	com.amazonaws. <i>region</i> .bedrock-agent
	com.amazonaws. <i>region</i> .bedrock-agent-runtime
	com.amazonaws. <i>region</i> .bedrock-runtime
AWS Billing Conductor	com.amazonaws. <i>region</i> .billingconductor

AWS service	Service name
Amazon Braket	com.amazonaws. <i>region</i> .braket
AWS Clean Rooms	com.amazonaws. <i>region</i> .cleanrooms
AWS Clean Rooms ML	com.amazonaws. <i>region</i> .cleanrooms-ml
AWS Cloud Control API	com.amazonaws. <i>region</i> .cloudcontrolapi
	com.amazonaws. <i>region</i> .cloudcontrolapi-fips
Amazon Cloud Directory	com.amazonaws. <i>region</i> .clouddirectory
AWS CloudFormation	com.amazonaws. <i>region</i> .cloudformation
AWS CloudHSM	com.amazonaws. <i>region</i> .cloudhsmv2
AWS Cloud Map	com.amazonaws. <i>region</i> .servicediscovery
	com.amazonaws. <i>region</i> .servicediscovery-fips
	com.amazonaws. <i>region</i> .data-servicediscovery
	com.amazonaws. <i>region</i> .data-servicediscovery-fips
AWS CloudTrail	com.amazonaws. <i>region</i> .cloudtrail
Amazon CloudWatch	com.amazonaws. <i>region</i> .evidently
	com.amazonaws. <i>region</i> .evidently-dataplane
	com.amazonaws. <i>region</i> .monitoring
	com.amazonaws. <i>region</i> .rum
	com.amazonaws. <i>region</i> .rum-dataplane
	com.amazonaws. <i>region</i> .synthetics
	com.amazonaws. <i>region</i> .events
Amazon CloudWatch Events	com.amazonaws. <i>region</i> .events

AWS service	Service name
Amazon CloudWatch Logs	com.amazonaws. <i>region</i> .logs
Amazon CloudWatch Network Monitor	com.amazonaws. <i>region</i> .networkmonitor
AWS CodeArtifact	com.amazonaws. <i>region</i> .codeartifact.api
	com.amazonaws. <i>region</i> .codeartifact.repositories
AWS CodeBuild	com.amazonaws. <i>region</i> .codebuild
	com.amazonaws. <i>region</i> .codebuild-fips
AWS CodeCommit	com.amazonaws. <i>region</i> .codecommit
	com.amazonaws. <i>region</i> .codecommit-fips
	com.amazonaws. <i>region</i> .git-codecommit
	com.amazonaws. <i>region</i> .git-codecommit-fips
AWS CodeDeploy	com.amazonaws. <i>region</i> .codedeploy
	com.amazonaws. <i>region</i> .codedeploy-commands-secure
Amazon CodeGuru Profiler	com.amazonaws. <i>region</i> .codeguru-profiler
Amazon CodeGuru Reviewer	com.amazonaws. <i>region</i> .codeguru-reviewer
AWS CodePipeline	com.amazonaws. <i>region</i> .codepipeline
AWS CodeStar Connections	com.amazonaws. <i>region</i> .codestar-connections.api
Amazon CodeWhisperer	com.amazonaws. <i>region</i> .codewhisperer
Amazon Comprehend	com.amazonaws. <i>region</i> .comprehend
Amazon Comprehend Medical	com.amazonaws. <i>region</i> .comprehendmedical
AWS Config	com.amazonaws. <i>region</i> .config

AWS service	Service name
Amazon Connect	com.amazonaws. <i>region</i> .app-integrations
	com.amazonaws. <i>region</i> .cases
	com.amazonaws. <i>region</i> .connect-campaigns
	com.amazonaws. <i>region</i> .profile
	com.amazonaws. <i>region</i> .voiceid
	com.amazonaws. <i>region</i> .wisdom
AWS Connector Service	com.amazonaws. <i>region</i> .awsconnector
AWS Control Catalog	com.amazonaws. <i>region</i> .controlcatalog
AWS Data Exchange	com.amazonaws. <i>region</i> .dataexchange
AWS Database Migration Service	com.amazonaws. <i>region</i> .dms
	com.amazonaws. <i>region</i> .dms-fips
AWS DataSync	com.amazonaws. <i>region</i> .datasync
Amazon DataZone	com.amazonaws. <i>region</i> .datazone
Amazon DevOps Guru	com.amazonaws. <i>region</i> .devops-guru
AWS Directory Service	com.amazonaws. <i>region</i> .ds
Amazon DynamoDB	com.amazonaws. <i>region</i> .dynamodb
Amazon EBS direct APIs	com.amazonaws. <i>region</i> .ebs
Amazon EC2	com.amazonaws. <i>region</i> .ec2
Amazon EC2 Auto Scaling	com.amazonaws. <i>region</i> .autoscaling
EC2 Image Builder	com.amazonaws. <i>region</i> .imagebuilder

AWS service	Service name
Amazon ECR	com.amazonaws. <i>region</i> .ecr.api
	com.amazonaws. <i>region</i> .ecr.dkr
Amazon ECS	com.amazonaws. <i>region</i> .ecs
	com.amazonaws. <i>region</i> .ecs-agent
	com.amazonaws. <i>region</i> .ecs-telemetry
Amazon EKS	com.amazonaws. <i>region</i> .eks
	com.amazonaws. <i>region</i> .eks-auth
AWS Elastic Beanstalk	com.amazonaws. <i>region</i> .elasticbeanstalk
	com.amazonaws. <i>region</i> .elasticbeanstalk-health
AWS Elastic Disaster Recovery	com.amazonaws. <i>region</i> .drs
Amazon Elastic File System	com.amazonaws. <i>region</i> .elasticfilesystem
	com.amazonaws. <i>region</i> .elasticfilesystem-fips
Amazon Elastic Inference	com.amazonaws. <i>region</i> .elastic-inference.runtime
Elastic Load Balancing	com.amazonaws. <i>region</i> .elasticloadbalancing
Amazon ElastiCache	com.amazonaws. <i>region</i> .elasticache
	com.amazonaws. <i>region</i> .elasticache-fips
AWS Elemental MediaConnect	com.amazonaws. <i>region</i> .mediaconnect
Amazon EMR	com.amazonaws. <i>region</i> .elasticmapreduce
Amazon EMR on EKS	com.amazonaws. <i>region</i> .emr-containers
Amazon EMR Serverless	com.amazonaws. <i>region</i> .emr-serverless

AWS service	Service name
Amazon EMR WAL	com.amazonaws. <i>region</i> .emrwal.prod
AWS Entity Resolution	com.amazonaws. <i>region</i> .entityresolution
Amazon EventBridge	com.amazonaws. <i>region</i> .events
AWS Fault Injection Service	com.amazonaws. <i>region</i> .fis
Amazon FinSpace	com.amazonaws. <i>region</i> .finspace
	com.amazonaws. <i>region</i> .finspace-api
Amazon Forecast	com.amazonaws. <i>region</i> .forecast
	com.amazonaws. <i>region</i> .forecastquery
	com.amazonaws. <i>region</i> .forecast-fips
	com.amazonaws. <i>region</i> .forecastquery-fips
Amazon Fraud Detector	com.amazonaws. <i>region</i> .frauddetector
Amazon FSx	com.amazonaws. <i>region</i> .fsx
	com.amazonaws. <i>region</i> .fsx-fips
AWS Glue	com.amazonaws. <i>region</i> .glue
AWS Glue DataBrew	com.amazonaws. <i>region</i> .databrew
Amazon Managed Grafana	com.amazonaws. <i>region</i> .grafana
	com.amazonaws. <i>region</i> .grafana-workspace
AWS Ground Station	com.amazonaws. <i>region</i> .groundstation
Amazon GuardDuty	com.amazonaws. <i>region</i> .guardduty-data
	com.amazonaws. <i>region</i> .guardduty-data-fips

AWS service	Service name
AWS HealthImaging	com.amazonaws. <i>region</i> .medical-imaging
	com.amazonaws. <i>region</i> .runtime-medical-imaging
AWS HealthLake	com.amazonaws. <i>region</i> .healthlake
IAM Identity Center	com.amazonaws. <i>region</i> .identitystore
IAM Roles Anywhere	com.amazonaws. <i>region</i> .rolesanywhere
Amazon Inspector	com.amazonaws. <i>region</i> .inspector2
AWS IoT Core	com.amazonaws. <i>region</i> .iot.data
	com.amazonaws. <i>region</i> .iot.credentials
	com.amazonaws. <i>region</i> .iot.fleethub.api
AWS IoT Core Device Advisor	com.amazonaws. <i>region</i> .deviceadvisor.iot
AWS IoT Core for LoRaWAN	com.amazonaws. <i>region</i> .iotwireless.api
	com.amazonaws. <i>region</i> .lorawan.cups
	com.amazonaws. <i>region</i> .lorawan.lns
AWS IoT FleetWise	com.amazonaws. <i>region</i> .iotfleetwise
AWS IoT Greengrass	com.amazonaws. <i>region</i> .greengrass
AWS IoT RoboRunner	com.amazonaws. <i>region</i> .iotroborunner
AWS IoT SiteWise	com.amazonaws. <i>region</i> .iotsitewise.api
	com.amazonaws. <i>region</i> .iotsitewise.data
AWS IoT TwinMaker	com.amazonaws. <i>region</i> .iottwinmaker.api
	com.amazonaws. <i>region</i> .iottwinmaker.data

AWS service	Service name
Amazon Kendra	com.amazonaws. <i>region</i> .kendra aws.api. <i>region</i> .kendra-ranking
AWS Key Management Service	com.amazonaws. <i>region</i> .kms com.amazonaws. <i>region</i> .kms-fips
Amazon Keyspaces (for Apache Cassandra)	com.amazonaws. <i>region</i> .cassandra com.amazonaws. <i>region</i> .cassandra-fips
Amazon Data Firehose	com.amazonaws. <i>region</i> .kinesis-firehose
Amazon Kinesis Data Streams	com.amazonaws. <i>region</i> .kinesis-streams
AWS Lake Formation	com.amazonaws. <i>region</i> .lakeformation
AWS Lambda	com.amazonaws. <i>region</i> .lambda
Amazon Lex	com.amazonaws. <i>region</i> .models-v2-lex com.amazonaws. <i>region</i> .runtime-v2-lex
AWS License Manager	com.amazonaws. <i>region</i> .license-manager com.amazonaws. <i>region</i> .license-manager-fips com.amazonaws. <i>region</i> .license-manager-user-subscriptions
Amazon Lookout for Equipment	com.amazonaws. <i>region</i> .lookoutequipment
Amazon Lookout for Metrics	com.amazonaws. <i>region</i> .lookoutmetrics
Amazon Lookout for Vision	com.amazonaws. <i>region</i> .lookoutvision
Amazon Macie	com.amazonaws. <i>region</i> .macie2
AWS Mainframe Modernization	com.amazonaws. <i>region</i> .m2

AWS service	Service name
Amazon Managed Blockchain	com.amazonaws. <i>region</i> .managedblockchain-query com.amazonaws. <i>region</i> .managedblockchain.bitcoin. mainnet com.amazonaws. <i>region</i> .managedblockchain.bitcoin. testnet
Amazon Managed Service for Prometheus	com.amazonaws. <i>region</i> .aps com.amazonaws. <i>region</i> .aps-workspaces
Amazon Managed Workflows for Apache Airflow	com.amazonaws. <i>region</i> .airflow.api com.amazonaws. <i>region</i> .airflow.env com.amazonaws. <i>region</i> .airflow.ops
AWS Management Console	com.amazonaws. <i>region</i> .console com.amazonaws. <i>region</i> .signin
Amazon MemoryDB for Redis	com.amazonaws. <i>region</i> .memory-db com.amazonaws. <i>region</i> .memorydb-fips
AWS Migration Hub Orchestrator	com.amazonaws. <i>region</i> .migrationhub-orchestrator
AWS Migration Hub Refactor Spaces	com.amazonaws. <i>region</i> .refactor-spaces
Migration Hub Strategy Recommendations	com.amazonaws. <i>region</i> .migrationhub-strategy
Amazon Neptune Analytics	com.amazonaws. <i>region</i> .neptune-graph
Amazon Nimble Studio	com.amazonaws. <i>region</i> .nimble
AWS HealthOmics	com.amazonaws. <i>region</i> .analytics-omics

AWS service	Service name
	com.amazonaws. <i>region</i> .control-storage-omics
	com.amazonaws. <i>region</i> .storage-omics
	com.amazonaws. <i>region</i> .tags-omics
	com.amazonaws. <i>region</i> .workflows-omics
Amazon OpenSearch Service	These endpoints are service-managed
AWS Organizations	com.amazonaws. <i>region</i> .organizations
	com.amazonaws. <i>region</i> .organizations-fips
AWS Panorama	com.amazonaws. <i>region</i> .panorama
AWS Payment Cryptography	com.amazonaws. <i>region</i> .payment-cryptography.contr olplane
	com.amazonaws. <i>region</i> .payment-cryptography.datap lane
Amazon Personalize	com.amazonaws. <i>region</i> .personalize
	com.amazonaws. <i>region</i> .personalize-events
	com.amazonaws. <i>region</i> .personalize-runtime
AWS Supply Chain	com.amazonaws. <i>region</i> .scn
Amazon Pinpoint	com.amazonaws. <i>region</i> .pinpoint
	com.amazonaws. <i>region</i> .pinpoint-sms-voice-v2
Amazon Polly	com.amazonaws. <i>region</i> .polly
AWS Private 5G	com.amazonaws. <i>region</i> .private-networks
AWS Private Certificate Authority	com.amazonaws. <i>region</i> .acm-pca

AWS service	Service name
	com.amazonaws. <i>region</i> .pca-connector-ad
AWS Proton	com.amazonaws. <i>region</i> .proton
Amazon QLDB	com.amazonaws. <i>region</i> .qldb.session
Amazon RDS	com.amazonaws. <i>region</i> .rds
Amazon RDS Data API	com.amazonaws. <i>region</i> .rds-data
AWS re:Post Private	com.amazonaws. <i>region</i> .repostspace
Amazon Redshift	com.amazonaws. <i>region</i> .redshift
	com.amazonaws. <i>region</i> .redshift-fips
Amazon Redshift Data API	com.amazonaws. <i>region</i> .redshift-data
Amazon Rekognition	com.amazonaws. <i>region</i> .rekognition
	com.amazonaws. <i>region</i> .rekognition-fips
	com.amazonaws. <i>region</i> .streaming-rekognition
	com.amazonaws. <i>region</i> .streaming-rekognition-fips
AWS RoboMaker	com.amazonaws. <i>region</i> .robomaker
Amazon S3	com.amazonaws. <i>region</i> .s3
Amazon S3 Multi-Region Access Points	com.amazonaws.s3-global.accesspoint
Amazon S3 on Outposts	com.amazonaws. <i>region</i> .s3-outposts
Amazon SageMaker	aws.sagemaker. <i>region</i> .notebook
	aws.sagemaker. <i>region</i> .studio
	com.amazonaws. <i>region</i> .sagemaker.api

AWS service	Service name
	com.amazonaws. <i>region</i> .sagemaker.featurestore-runtime
	com.amazonaws. <i>region</i> .sagemaker.metrics
	com.amazonaws. <i>region</i> .sagemaker.runtime
	com.amazonaws. <i>region</i> .sagemaker.runtime-fips
AWS Secrets Manager	com.amazonaws. <i>region</i> .secretsmanager
AWS Security Hub	com.amazonaws. <i>region</i> .securityhub
AWS Security Token Service	com.amazonaws. <i>region</i> .sts
Service Catalog	com.amazonaws. <i>region</i> .servicecatalog
	com.amazonaws. <i>region</i> .servicecatalog-appregistry
Amazon SES	com.amazonaws. <i>region</i> .email-smtp
AWS SimSpace Weaver	com.amazonaws. <i>region</i> .simspaceweaver
AWS Snow Device Management	com.amazonaws. <i>region</i> .snow-device-management
Amazon SNS	com.amazonaws. <i>region</i> .sns
Amazon SQS	com.amazonaws. <i>region</i> .sqs
Amazon SWF	com.amazonaws. <i>region</i> .swf
	com.amazonaws. <i>region</i> .swf-fips
AWS Step Functions	com.amazonaws. <i>region</i> .states
	com.amazonaws. <i>region</i> .sync-states
AWS Storage Gateway	com.amazonaws. <i>region</i> .storagegateway
AWS Systems Manager	com.amazonaws. <i>region</i> .ec2messages

AWS service	Service name
	com.amazonaws. <i>region</i> .ssm
	com.amazonaws. <i>region</i> .ssm-contacts
	com.amazonaws. <i>region</i> .ssm-incidents
	com.amazonaws. <i>region</i> .ssmmessages
AWS Telco Network Builder	com.amazonaws. <i>region</i> .tnb
Amazon Textract	com.amazonaws. <i>region</i> .textract
	com.amazonaws. <i>region</i> .textract-fips
Amazon Timestream	com.amazonaws. <i>region</i> .timestream.ingest- <i>cell</i>
	com.amazonaws. <i>region</i> .timestream.query- <i>cell</i>
Amazon Timestream for InfluxDB	com.amazonaws. <i>region</i> .timestream-influxdb
Amazon Transcribe	com.amazonaws. <i>region</i> .transcribe
	com.amazonaws. <i>region</i> .transcribestreaming
Amazon Transcribe Medical	com.amazonaws. <i>region</i> .transcribe
	com.amazonaws. <i>region</i> .transcribestreaming
AWS Transfer for SFTP	com.amazonaws. <i>region</i> .transfer
	com.amazonaws. <i>region</i> .transfer.server
Amazon Translate	com.amazonaws. <i>region</i> .translate
AWS Trusted Advisor	com.amazonaws. <i>region</i> .trustedadvisor
Amazon Verified Permissions	com.amazonaws. <i>region</i> .verifiedpermissions
Amazon VPC Lattice	com.amazonaws. <i>region</i> .vpc-lattice

AWS service	Service name
Amazon WorkSpaces	com.amazonaws. <i>region</i> .workspaces
Amazon WorkSpaces Thin Client	com.amazonaws. <i>region</i> .thinclient.api
AWS X-Ray	com.amazonaws. <i>region</i> .xray

View available AWS service names

You can use the [describe-vpc-endpoint-services](#) command to view the service names that support VPC endpoints.

The following example displays the AWS services that support interface endpoints in the specified Region. The `--query` option limits the output to the service names.

```
aws ec2 describe-vpc-endpoint-services \
  --filters Name=service-type,Values=Interface Name=owner,Values=amazon \
  --region us-east-1 \
  --query ServiceNames
```

The following is example output:

```
[
  "aws.api.us-east-1.kendra-ranking",
  "aws.sagemaker.us-east-1.notebook",
  "aws.sagemaker.us-east-1.studio",
  "com.amazonaws.s3-global.accesspoint",
  "com.amazonaws.us-east-1.access-analyzer",
  "com.amazonaws.us-east-1.account",
  ...
]
```

View information about a service

After you have the service name, you can use the [describe-vpc-endpoint-services](#) command to view detailed information about each endpoint service.

The following example displays information about the Amazon CloudWatch interface endpoint in the specified Region.

```
aws ec2 describe-vpc-endpoint-services \
  --service-name "com.amazonaws.us-east-1.monitoring" \
  --region us-east-1
```

The following is example output. `VpcEndpointPolicySupported` indicates whether [endpoint policies](#) are supported. `SupportedIpAddressTypes` indicates which IP address types are supported .

```
{
  "ServiceDetails": [
    {
      "ServiceName": "com.amazonaws.us-east-1.monitoring",
      "ServiceId": "vpce-svc-0fc975f3e7e5beba4",
      "ServiceType": [
        {
          "ServiceType": "Interface"
        }
      ],
      "AvailabilityZones": [
        "us-east-1a",
        "us-east-1b",
        "us-east-1c",
        "us-east-1d",
        "us-east-1e",
        "us-east-1f"
      ],
      "Owner": "amazon",
      "BaseEndpointDnsNames": [
        "monitoring.us-east-1.vpce.amazonaws.com"
      ],
      "PrivateDnsName": "monitoring.us-east-1.amazonaws.com",
      "PrivateDnsNames": [
        {
          "PrivateDnsName": "monitoring.us-east-1.amazonaws.com"
        }
      ],
      "VpcEndpointPolicySupported": true,
      "AcceptanceRequired": false,
      "ManagesVpcEndpoints": false,
      "Tags": [],
      "PrivateDnsNameVerificationState": "verified",
      "SupportedIpAddressTypes": [
```

```

        "ipv4"
      ]
    }
  ],
  "ServiceNames": [
    "com.amazonaws.us-east-1.monitoring"
  ]
}

```

View endpoint policy support

To verify whether a service supports [endpoint policies](#), call the [describe-vpc-endpoint-services](#) command and check the value of `VpcEndpointPolicySupported`. The possible values are `true` and `false`.

The following example checks whether the specified service supports endpoint policies in the specified Region. The `--query` option limits the output to the value of `VpcEndpointPolicySupported`.

```

aws ec2 describe-vpc-endpoint-services \
  --service-name "com.amazonaws.us-east-1.s3" \
  --region us-east-1 \
  --query ServiceDetails[*].VpcEndpointPolicySupported \
  --output text

```

The following is example output.

```
True
```

The following example lists the AWS services that support endpoint policies in the specified Region. The `--query` option limits the output to the service names. To run this command using the Windows command prompt, remove the single quotes around the query string, and change the line continuation character from `\` to `^`.

```

aws ec2 describe-vpc-endpoint-services \
  --filters Name=service-type,Values=Interface Name=owner,Values=amazon \
  --region us-east-1 \
  --query 'ServiceDetails[?VpcEndpointPolicySupported==`true`].ServiceName'

```

The following is example output.

```
[
  "aws.api.us-east-1.kendra-ranking",
  "aws.sagemaker.us-east-1.notebook",
  "aws.sagemaker.us-east-1.studio",
  "com.amazonaws.s3-global.accesspoint",
  "com.amazonaws.us-east-1.access-analyzer",
  "com.amazonaws.us-east-1.account",
  ...
]
```

The following example lists the AWS services that do not support endpoint policies in the specified Region. The `--query` option limits the output to the service names. To run this command using the Windows command prompt, remove the single quotes around the query string, and change the line continuation character from `\` to `^`.

```
aws ec2 describe-vpc-endpoint-services \
  --filters Name=service-type,Values=Interface Name=owner,Values=amazon \
  --region us-east-1 \
  --query 'ServiceDetails[?VpcEndpointPolicySupported==`false`].ServiceName'
```

The following is example output.

```
[
  "com.amazonaws.us-east-1.appmesh-envoy-management",
  "com.amazonaws.us-east-1.apprunner.requests",
  "com.amazonaws.us-east-1.appstream.api",
  "com.amazonaws.us-east-1.appstream.streaming",
  "com.amazonaws.us-east-1.awsconnector",
  "com.amazonaws.us-east-1.cleanrooms",
  "com.amazonaws.us-east-1.cloudtrail",
  "com.amazonaws.us-east-1.codeguru-profiler",
  "com.amazonaws.us-east-1.codeguru-reviewer",
  "com.amazonaws.us-east-1.codepipeline",
  "com.amazonaws.us-east-1.codewhisperer",
  "com.amazonaws.us-east-1.datasync",
  "com.amazonaws.us-east-1.datazone",
  "com.amazonaws.us-east-1.deviceadvisor.iot",
  "com.amazonaws.us-east-1.ebs",
  "com.amazonaws.us-east-1.eks",
  "com.amazonaws.us-east-1.elastic-inference.runtime",
  "com.amazonaws.us-east-1.email-smtp",
  "com.amazonaws.us-east-1.grafana-workspace",

```

```

"com.amazonaws.us-east-1.iot.credentials",
"com.amazonaws.us-east-1.iot.data",
"com.amazonaws.us-east-1.iotwireless.api",
"com.amazonaws.us-east-1.lorawan.cups",
"com.amazonaws.us-east-1.lorawan.lns",
"com.amazonaws.us-east-1.macie2",
"com.amazonaws.us-east-1.neptune-graph",
"com.amazonaws.us-east-1.nimble",
"com.amazonaws.us-east-1.organizations",
"com.amazonaws.us-east-1.redshift-data",
"com.amazonaws.us-east-1.refactor-spaces",
"com.amazonaws.us-east-1.sagemaker.runtime-fips",
"com.amazonaws.us-east-1.storagegateway",
"com.amazonaws.us-east-1.transfer",
"com.amazonaws.us-east-1.transfer.server",
"com.amazonaws.us-east-1.verifiedpermissions"
]

```

View IPv6 support

You can use the following [describe-vpc-endpoint-services](#) command to view the AWS services that you can access over IPv6 in the specified Region. The `--query` option limits the output to the service names.

```

aws ec2 describe-vpc-endpoint-services \
  --filters Name=supported-ip-address-types,Values=ipv6 Name=owner,Values=amazon
  Name=service-type,Values=Interface \
  --region us-east-1 \
  --query ServiceNames

```

The following is example output:

```

[
  "aws.api.us-east-1.kendra-ranking",
  "com.amazonaws.us-east-1.athena",
  "com.amazonaws.us-east-1.data-servicediscovery",
  "com.amazonaws.us-east-1.data-servicediscovery-fips",
  "com.amazonaws.us-east-1.eks-auth",
  "com.amazonaws.us-east-1.glue",
  "com.amazonaws.us-east-1.lakeformation",
  "com.amazonaws.us-east-1.s3-outposts",
  "com.amazonaws.us-east-1.servicediscovery",

```

```
"com.amazonaws.us-east-1.servicediscovery-fips",  
"com.amazonaws.us-east-1.timestream-influxdb"  
]
```

Access an AWS service using an interface VPC endpoint

You can create an interface VPC endpoint to connect to services powered by AWS PrivateLink, including many AWS services. For an overview, see [the section called “Concepts”](#) and [Access AWS services](#).

For each subnet that you specify from your VPC, we create an endpoint network interface in the subnet and assign it a private IP address from the subnet address range. An endpoint network interface is a requester-managed network interface; you can view it in your AWS account, but you can't manage it yourself.

You are billed for hourly usage and data processing charges. For more information, see [Interface endpoint pricing](#).

Contents

- [Prerequisites](#)
- [Create a VPC endpoint](#)
- [Shared subnets](#)

Prerequisites

- Deploy the resources that will access the AWS service in your VPC.
- To use private DNS, you must enable DNS hostnames and DNS resolution for your VPC. For more information, see [View and update DNS attributes](#) in the *Amazon VPC User Guide*.
- To enable IPv6 for an interface endpoint, the AWS service must support access over IPv6. For more information, see [the section called “IP address types”](#).
- Create a security group that allows the resources in your VPC to communicate with the endpoint network interfaces for the VPC endpoint. To ensure that tools such as the AWS CLI can make requests over HTTPS from resources in the VPC to the AWS service, the security group must allow inbound HTTPS traffic.
- If your resources are in a subnet with a network ACL, verify that the network ACL allows traffic between the endpoint network interfaces and the resources in the VPC.

- There are quotas on your AWS PrivateLink resources. For more information, see [AWS PrivateLink quotas](#).

Create a VPC endpoint

Use the following procedure to create an interface VPC endpoint that connects to an AWS service.

To create an interface endpoint for an AWS service

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create endpoint**.
4. For **Service category**, choose **AWS services**.
5. For **Service name**, select the service. For more information, see [the section called "Services that integrate"](#).
6. For **VPC**, select the VPC from which you'll access the AWS service.
7. If, in Step 5, you selected the service name for Amazon S3, and if you want to configure [private DNS support](#), select **Additional settings, Enable DNS name**. When you make this selection, it also automatically selects **Enable private DNS only for inbound endpoint**. You can configure private DNS with an inbound Resolver endpoint only for interface endpoints for Amazon S3. If you do not have a gateway endpoint for Amazon S3 and you select **Enable private DNS only for inbound endpoint**, you'll receive an error when you attempt the final step in this procedure.

If, in Step 5, you selected the service name for any service other than Amazon S3, **Additional settings, Enable DNS name** is already selected. We recommend that you keep the default.

8. For **Subnets**, select one subnet per Availability Zone from which you'll access the AWS service. You can't select multiple subnets from the same Availability Zone. We create an endpoint network interface in each subnet that you select. By default, we select IP addresses from the subnet IP address ranges and assign them to the endpoint network interfaces. To choose the IP addresses for an endpoint network interface, select **Designate IP addresses** and enter an IPv4 address from the subnet address range. If the endpoint service supports IPv6, you can also enter an IPv6 address from the subnet address range.
9. For **IP address type**, choose from the following options:

- **IPv4** – Assign IPv4 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have IPv4 address ranges and the service accepts IPv4 requests.
 - **IPv6** – Assign IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets are IPv6 only subnets and the service accepts IPv6 requests.
 - **Dualstack** – Assign both IPv4 and IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have both IPv4 and IPv6 address ranges and the service accepts both IPv4 and IPv6 requests.
10. For **Security groups**, select the security groups to associate with the endpoint network interfaces for the VPC endpoint. By default, we associate the default security group for the VPC.
 11. For **Policy**, select **Full access** to allow all operations by all principals on all resources over the VPC endpoint. Otherwise, select **Custom** to attach a VPC endpoint policy that controls the permissions that principals have for performing actions on resources over the VPC endpoint. This option is available only if the service supports VPC endpoint policies. For more information, see [Endpoint policies](#).
 12. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
 13. Choose **Create endpoint**.

To create an interface endpoint using the command line

- [create-vpc-endpoint](#) (AWS CLI)
- [New-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Shared subnets

You can't create, describe, modify, or delete VPC endpoints in subnets that are shared with you. However, you can use the VPC endpoints in subnets that are shared with you.

Configure an interface endpoint

After you create an interface VPC endpoint, you can update its configuration.

Tasks

- [Add or remove subnets](#)

- [Associate security groups](#)
- [Edit the VPC endpoint policy](#)
- [Enable private DNS names](#)
- [Manage tags](#)

Add or remove subnets

You can choose one subnet per Availability Zone for your interface endpoint. If you add a subnet, we create an endpoint network interface in the subnet and assign it a private IP address from the IP address range of the subnet. If you remove a subnet, we delete its endpoint network interface. For more information, see [the section called “Subnets and Availability Zones”](#).

To change the subnets using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the interface endpoint.
4. Choose **Actions, Manage subnets**.
5. Select or deselect Availability Zones as needed. For each Availability Zone, select one subnet. By default, we select IP addresses from the subnet IP address ranges and assign them to the endpoint network interfaces. To choose the IP addresses for an endpoint network interface, select **Designate IP addresses** and enter an IPv4 address from the subnet address range. If the endpoint service supports IPv6, you can also enter an IPv6 address from the subnet address range.

If you specify an IP address for a subnet that already has an endpoint network interface for this VPC endpoint, we replace the endpoint network interface with a new one. This process temporarily disconnects the subnet and the VPC endpoint.

6. Choose **Modify subnets**.

To change the subnets using the command line

- [modify-vpc-endpoint](#) (AWS CLI)
- [Edit-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Associate security groups

You can change the security groups that are associated with the network interfaces for your interface endpoint. The security group rules control the traffic that is allowed to the endpoint network interface from the resources in your VPC.

To change the security groups using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the interface endpoint.
4. Choose **Actions, Manage security groups**.
5. Select or deselect security groups as needed.
6. Choose **Modify security groups**.

To change the security groups using the command line

- [modify-vpc-endpoint](#) (AWS CLI)
- [Edit-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Edit the VPC endpoint policy

If the AWS service supports endpoint policies you can edit the endpoint policy for the endpoint. After you update an endpoint policy, it can take a few minutes for the changes to take effect. For more information, see [Endpoint policies](#).

To change the endpoint policy using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the interface endpoint.
4. Choose **Actions, Manage policy**.
5. Choose **Full Access** to allow full access to the service, or choose **Custom** and attach a custom policy.
6. Choose **Save**.

To change the endpoint policy using the command line

- [modify-vpc-endpoint](#) (AWS CLI)
- [Edit-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Enable private DNS names

You can enable private DNS names for your VPC endpoint. To use private DNS names, you must enable both [DNS hostnames and DNS resolution](#) for your VPC. After you enable private DNS names, it might take a few minutes for the private IP addresses to become available. The DNS records that we create when you enable private DNS names are private. Therefore, the private DNS name is not publicly resolvable.

To change the private DNS names option using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the interface endpoint.
4. Choose **Actions, Modify private DNS name**.
5. Select or clear **Enable for this endpoint** as required.
6. If the service is Amazon S3, selecting **Enable for this endpoint** in the previous step also selects **Enable private DNS only for inbound endpoint**. If you prefer the standard private DNS functionality, clear **Enable private DNS only for inbound endpoint**. If you do not have a gateway endpoint for Amazon S3 in addition to an interface endpoint for Amazon S3, and you select **Enable private DNS only for inbound endpoint**, you'll receive an error when you save changes in the next step. For more information, see [the section called "Private DNS"](#).
7. Choose **Save changes**.

To change the private DNS names option using the command line

- [modify-vpc-endpoint](#) (AWS CLI)
- [Edit-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Manage tags

You can tag your interface endpoint to help you identify it or categorize it according to your organization's needs.

To manage tags using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the interface endpoint.
4. Choose **Actions, Manage tags**.
5. For each tag to add choose **Add new tag** and enter the tag key and tag value.
6. To remove a tag, choose **Remove** to the right of the tag key and value.
7. Choose **Save**.

To manage tags using the command line

- [create-tags](#) and [delete-tags](#) (AWS CLI)
- [New-EC2Tag](#) and [Remove-EC2Tag](#) (Tools for Windows PowerShell)

Receive alerts for interface endpoint events

You can create a notification to receive alerts for specific events related to your interface endpoint. For example, you can receive an email when a connection request is accepted or rejected.

Tasks

- [Create an SNS notification](#)
- [Add an access policy](#)
- [Add a key policy](#)

Create an SNS notification

Use the following procedure to create an Amazon SNS topic for the notifications and subscribe to the topic.

To create a notification for an interface endpoint using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the interface endpoint.
4. From the **Notifications** tab, choose **Create notification**.
5. For **Notification ARN**, choose the ARN for the SNS topic that you created.
6. To subscribe to an event, select it from **Events**.
 - **Connect** – The service consumer created the interface endpoint. This sends a connection request to the service provider.
 - **Accept** – The service provider accepted the connection request.
 - **Reject** – The service provider rejected the connection request.
 - **Delete** – The service consumer deleted the interface endpoint.
7. Choose **Create notification**.

To create a notification for an interface endpoint using the command line

- [create-vpc-endpoint-connection-notification](#) (AWS CLI)
- [New-EC2VpcEndpointConnectionNotification](#) (Tools for Windows PowerShell)

Add an access policy

Add an access policy to the Amazon SNS topic that allows AWS PrivateLink to publish notifications on your behalf, such as the following. For more information, see [How do I edit my Amazon SNS topic's access policy?](#) Use the `aws:SourceArn` and `aws:SourceAccount` global condition keys to protect against the [confused deputy problem](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vpce.amazonaws.com"
      },
      "Action": "SNS:Publish",
```

```

    "Resource": "arn:aws:sns:region:account-id:topic-name",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:ec2:region:account-id:vpc-endpoint/endpoint-id"
      },
      "StringEquals": {
        "aws:SourceAccount": "account-id"
      }
    }
  }
]
}

```

Add a key policy

If you're using encrypted SNS topics, the resource policy for the KMS key must trust AWS PrivateLink to call AWS KMS API operations. The following is an example key policy.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vpce.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:region:account-id:key/key-id",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:ec2:region:account-id:vpc-endpoint/endpoint-id"
        },
        "StringEquals": {
          "aws:SourceAccount": "account-id"
        }
      }
    }
  ]
}

```

Delete an interface endpoint

When you are finished with a VPC endpoint, you can delete it. Deleting an interface endpoint also deletes its endpoint network interfaces.

To delete an interface endpoint using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the interface endpoint.
4. Choose **Actions, Delete VPC endpoints**.
5. When prompted for confirmation, enter **delete**.
6. Choose **Delete**.

To delete an interface endpoint using the command line

- [delete-vpc-endpoints](#) (AWS CLI)
- [Remove-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Gateway endpoints

Gateway VPC endpoints provide reliable connectivity to Amazon S3 and DynamoDB without requiring an internet gateway or a NAT device for your VPC. Gateway endpoints do not use AWS PrivateLink, unlike other types of VPC endpoints.

Amazon S3 and DynamoDB support both gateway endpoints and interface endpoints. For a comparison of the options, see the following:

- [Types of VPC endpoints for Amazon S3](#)
- [Types of VPC endpoints for Amazon DynamoDB](#)

Pricing

There is no additional charge for using gateway endpoints.

Contents

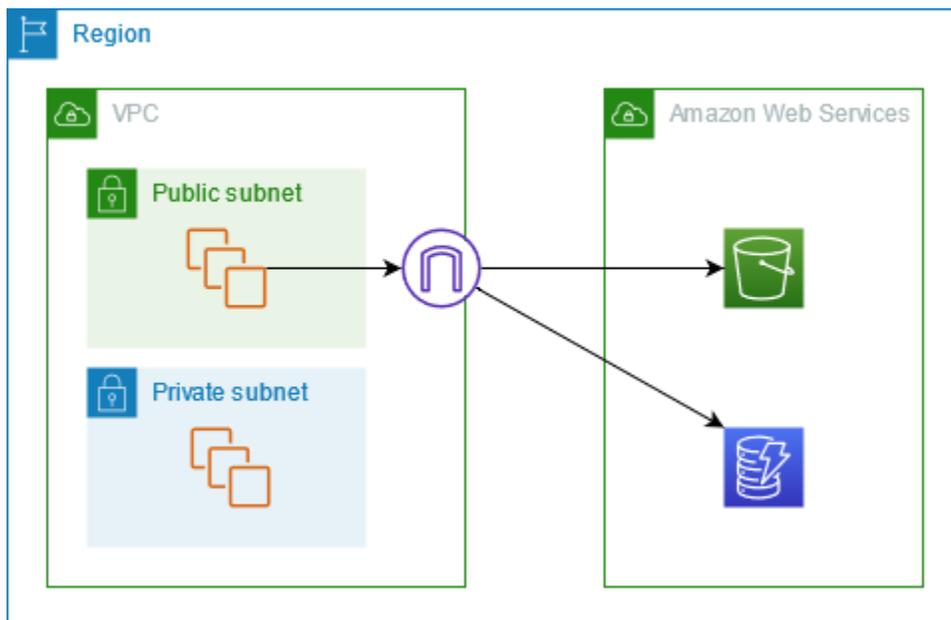
- [Overview](#)
- [Routing](#)
- [Security](#)
- [Gateway endpoints for Amazon S3](#)
- [Gateway endpoints for Amazon DynamoDB](#)

Overview

You can access Amazon S3 and DynamoDB through their public service endpoints or through gateway endpoints. This overview compares these methods.

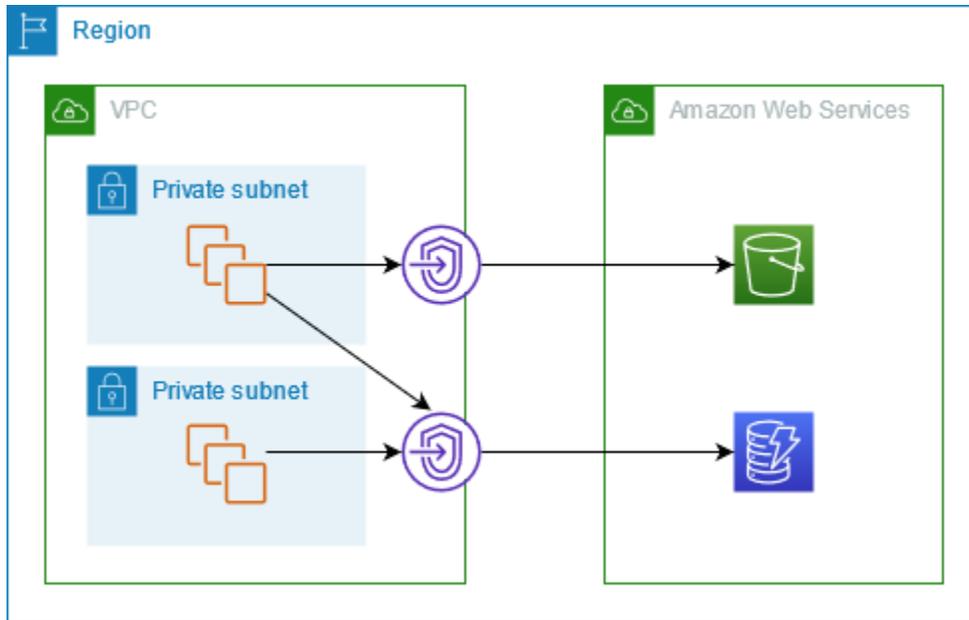
Access through an internet gateway

The following diagram shows how instances access Amazon S3 and DynamoDB through their public service endpoints. Traffic to Amazon S3 or DynamoDB from an instance in a public subnet is routed to the internet gateway for the VPC and then to the service. Instances in a private subnet can't send traffic to Amazon S3 or DynamoDB, because by definition private subnets do not have routes to an internet gateway. To enable instances in the private subnet to send traffic to Amazon S3 or DynamoDB, you would add a NAT device to the public subnet and route traffic in the private subnet to the NAT device. While traffic to Amazon S3 or DynamoDB traverses the internet gateway, it does not leave the AWS network.



Access through a gateway endpoint

The following diagram shows how instances access Amazon S3 and DynamoDB through a gateway endpoint. Traffic from your VPC to Amazon S3 or DynamoDB is routed to the gateway endpoint. Each subnet route table must have a route that sends traffic destined for the service to the gateway endpoint using the prefix list for the service. For more information, see [AWS-managed prefix lists](#) in the *Amazon VPC User Guide*.



Routing

When you create a gateway endpoint, you select the VPC route tables for the subnets that you enable. The following route is automatically added to each route table that you select. The destination is a prefix list for the service owned by AWS and the target is the gateway endpoint.

Destination	Target
<i>prefix_list_id</i>	<i>gateway_endpoint_id</i>

Considerations

- You can review the endpoint routes that we add to your route table, but you cannot modify or delete them. To add an endpoint route to a route table, associate it with the gateway endpoint. We delete the endpoint route when you disassociate the route table from the gateway endpoint or when you delete the gateway endpoint.

- All instances in the subnets associated with a route table associated with a gateway endpoint automatically use the gateway endpoint to access the service. Instances in subnets that aren't associated with these route tables use the public service endpoint, not the gateway endpoint.
- A route table can have both an endpoint route to Amazon S3 and an endpoint route to DynamoDB. You can have endpoint routes to the same service (Amazon S3 or DynamoDB) in multiple route tables. You can't have multiple endpoint routes to the same service (Amazon S3 or DynamoDB) in a single route table.
- We use the most specific route that matches the traffic to determine how to route the traffic (longest prefix match). For route tables with an endpoint route, this means the following:
 - If there is a route that sends all internet traffic (0.0.0.0/0) to an internet gateway, the endpoint route takes precedence for traffic destined for the service (Amazon S3 or DynamoDB) in the current Region. Traffic destined for a different AWS service uses the internet gateway.
 - Traffic that's destined for the service (Amazon S3 or DynamoDB) in a different Region goes to the internet gateway because prefix lists are specific to a Region.
 - If there is a route that specifies the exact IP address range for the service (Amazon S3 or DynamoDB) in the same Region, that route takes precedence over the endpoint route.

Security

When your instances access Amazon S3 or DynamoDB through a gateway endpoint, they access the service using its public endpoint. The security groups for these instances must allow traffic to and from the service. The following is an example outbound rule. It references the ID of the [prefix list](#) for the service.

Destination	Protocol	Port range
<i>prefix_list_id</i>	TCP	443

The network ACLs for the subnets for these instances must also allow traffic to and from the service. The following is an example outbound rule. You can't reference prefix lists in network ACL rules, but you can get the IP address ranges for the service from its prefix list.

Destination	Protocol	Port range
<i>service_cidr_block_1</i>	TCP	443

Destination	Protocol	Port range
<code>service_cidr_block_2</code>	TCP	443
<code>service_cidr_block_3</code>	TCP	443

Gateway endpoints for Amazon S3

You can access Amazon S3 from your VPC using gateway VPC endpoints. After you create the gateway endpoint, you can add it as a target in your route table for traffic destined from your VPC to Amazon S3.

There is no additional charge for using gateway endpoints.

Amazon S3 supports both gateway endpoints and interface endpoints. With a gateway endpoint, you can access Amazon S3 from your VPC, without requiring an internet gateway or NAT device for your VPC, and with no additional cost. However, gateway endpoints do not allow access from on-premises networks, from peered VPCs in other AWS Regions, or through a transit gateway. For those scenarios, you must use an interface endpoint, which is available for an additional cost. For more information, see [Types of VPC endpoints for Amazon S3](#) in the *Amazon S3 User Guide*.

Contents

- [Considerations](#)
- [Private DNS](#)
- [Create a gateway endpoint](#)
- [Control access using bucket policies](#)
- [Associate route tables](#)
- [Edit the VPC endpoint policy](#)
- [Delete a gateway endpoint](#)

Considerations

- A gateway endpoint is available only in the Region where you created it. Be sure to create your gateway endpoint in the same Region as your S3 buckets.

- If you're using the Amazon DNS servers, you must enable both [DNS hostnames and DNS resolution](#) for your VPC. If you're using your own DNS server, ensure that requests to Amazon S3 resolve correctly to the IP addresses maintained by AWS.
- The rules for the security groups for your instances that access Amazon S3 through a gateway endpoint must allow traffic to and from Amazon S3. You can reference the ID of the [prefix list](#) for Amazon S3 in security group rules.
- The network ACL for the subnet for your instances that access Amazon S3 through a gateway endpoint must allow traffic to and from Amazon S3. You can't reference prefix lists in network ACL rules, but you can get the IP address range for Amazon S3 from the [prefix list](#) for Amazon S3.
- Check whether you are using an AWS service that requires access to an S3 bucket. For example, a service might require access to buckets that contain log files, or might require you to download drivers or agents to your EC2 instances. If so, ensure that your endpoint policy allows the AWS service or resource to access these buckets using the `s3:GetObject` action.
- You can't use the `aws:SourceIp` condition in an identity policy or a bucket policy for requests to Amazon S3 that traverse a VPC endpoint. Instead, use the `aws:VpcSourceIp` condition. Alternatively, you can use route tables to control which EC2 instances can access Amazon S3 through the VPC endpoint.
- Gateway endpoints support only IPv4 traffic.
- The source IPv4 addresses from instances in your affected subnets as received by Amazon S3 change from public IPv4 addresses to the private IPv4 addresses in your VPC. An endpoint switches network routes, and disconnects open TCP connections. The previous connections that used public IPv4 addresses are not resumed. We recommend that you do not have any critical tasks running when you create or modify an endpoint; or that you test to ensure that your software can automatically reconnect to Amazon S3 after the connection break.
- Endpoint connections cannot be extended out of a VPC. Resources on the other side of a VPN connection, VPC peering connection, transit gateway, or AWS Direct Connect connection in your VPC cannot use a gateway endpoint to communicate with Amazon S3.
- Your account has a default quota of 20 gateway endpoints per Region, which is adjustable. There is also a limit of 255 gateway endpoints per VPC.

Private DNS

You can configure private DNS to optimize costs when you create both a gateway endpoint and an interface endpoint for Amazon S3.

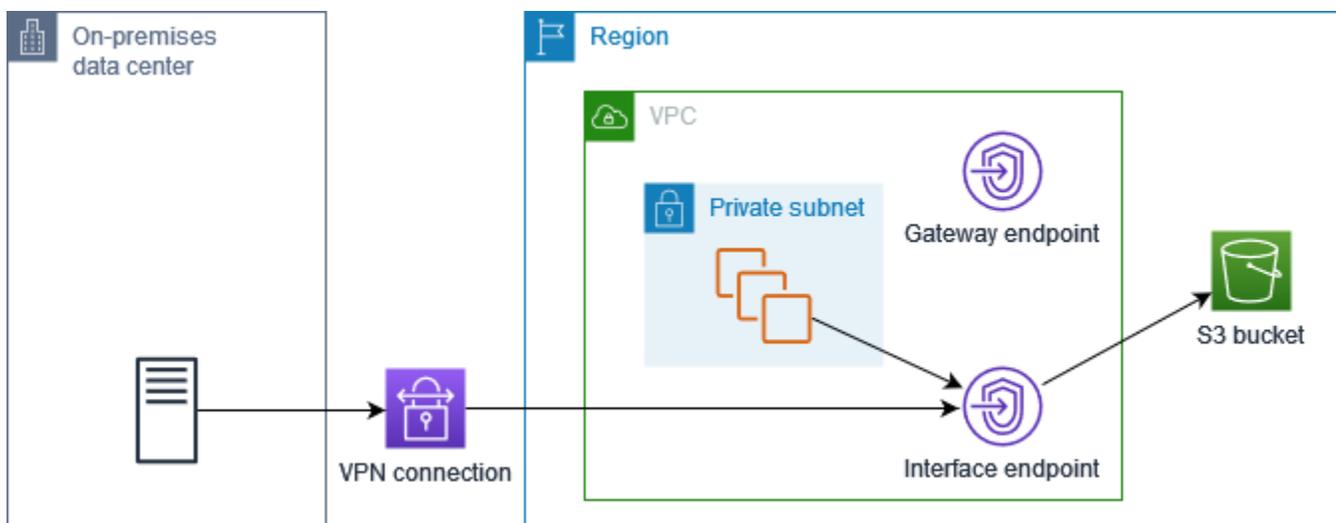
Route 53 Resolver

Amazon provides a DNS server, called the [Route 53 Resolver](#), for your VPC. The Route 53 Resolver automatically resolves local VPC domain names and records in private hosted zones. However, you can't use the Route 53 Resolver from outside your VPC. Route 53 provides Resolver endpoints and Resolver rules so that you can use the Route 53 Resolver from outside your VPC. An *inbound Resolver endpoint* forwards DNS queries from the on-premises network to Route 53 Resolver. An *outbound Resolver endpoint* forwards DNS queries from the Route 53 Resolver to the on-premises network.

When you configure your interface endpoint for Amazon S3 to use private DNS only for the inbound Resolver endpoint, we create an inbound Resolver endpoint. The inbound Resolver endpoint resolves DNS queries to Amazon S3 from on-premises to the private IP addresses of the interface endpoint. We also add ALIAS records for the Route 53 Resolver to the public hosted zone for Amazon S3, so that DNS queries from your VPC resolve to the Amazon S3 public IP addresses, which routes traffic to the gateway endpoint.

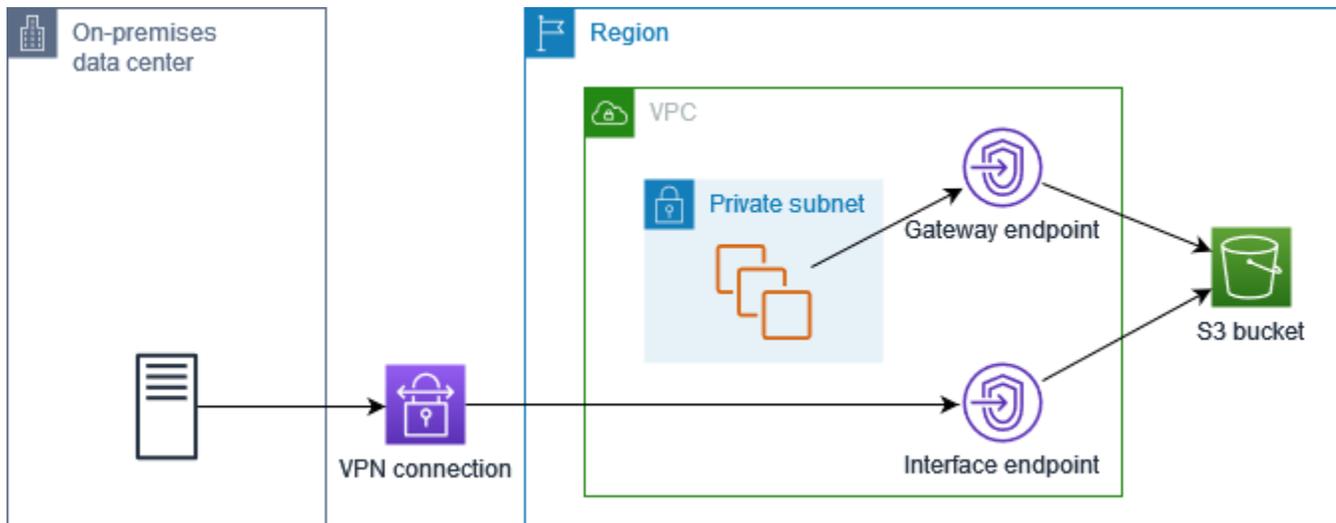
Private DNS

If you configure private DNS for your interface endpoint for Amazon S3 but do not configure private DNS only for the inbound Resolver endpoint, requests from both your on-premises network and your VPC use the interface endpoint to access Amazon S3. Therefore, you pay to use the interface endpoint for traffic from the VPC, instead of using the gateway endpoint for no additional charge.



Private DNS only for the inbound Resolver endpoint

If you configure private DNS only for the inbound Resolver endpoint, requests from your on-premises network use the interface endpoint to access Amazon S3, and requests from your VPC use the gateway endpoint to access Amazon S3. Therefore, you optimize your costs, because you pay to use the interface endpoint only for traffic that can't use the gateway endpoint.



Configure private DNS

You can configure private DNS for an interface endpoint for Amazon S3 when you create it or after you create it. For more information, see [the section called “Create a VPC endpoint”](#) (configure during creation) or [the section called “Enable private DNS names”](#) (configure after creation).

Create a gateway endpoint

Use the following procedure to create a gateway endpoint that connects to Amazon S3.

To create a gateway endpoint using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create endpoint**.
4. For **Service category**, choose **AWS services**.
5. For **Services**, add the filter **Type: Gateway** and select **com.amazonaws.region.s3**.
6. For **VPC**, select the VPC in which to create the endpoint.
7. For **Route tables**, select the route tables to be used by the endpoint. We automatically add a route that points traffic destined for the service to the endpoint network interface.

8. For **Policy**, select **Full access** to allow all operations by all principals on all resources over the VPC endpoint. Otherwise, select **Custom** to attach a VPC endpoint policy that controls the permissions that principals have to perform actions on resources over the VPC endpoint.
9. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
10. Choose **Create endpoint**.

To create a gateway endpoint using the command line

- [create-vpc-endpoint](#) (AWS CLI)
- [New-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Control access using bucket policies

You can use bucket policies to control access to buckets from specific endpoints, VPCs, IP address ranges, and AWS accounts. These examples assume that there are also policy statements that allow the access required for your use cases.

Example Example: Restrict access to a specific endpoint

You can create a bucket policy that restricts access to a specific endpoint by using the [aws:sourceVpce](#) condition key. The following policy denies access to the specified bucket using the specified actions unless the specified gateway endpoint is used. Note that this policy blocks access to the specified bucket using the specified actions through the AWS Management Console.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow-access-to-specific-VPCE",
      "Effect": "Deny",
      "Principal": "*",
      "Action": ["s3:PutObject", "s3:GetObject", "s3:DeleteObject"],
      "Resource": ["arn:aws:s3:::bucket_name",
                  "arn:aws:s3:::bucket_name/*"],
      "Condition": {
        "StringNotEquals": {
          "aws:sourceVpce": "vpce-1a2b3c4d"
        }
      }
    }
  ]
}
```

```
]
}
```

Example Example: Restrict access to a specific VPC

You can create a bucket policy that restricts access to specific VPCs by using the [aws:sourceVpc](#) condition key. This is useful if you have multiple endpoints configured in the same VPC. The following policy denies access to the specified bucket using the specified actions unless the request comes from the specified VPC. Note that this policy blocks access to the specified bucket using the specified actions through the AWS Management Console.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow-access-to-specific-VPC",
      "Effect": "Deny",
      "Principal": "*",
      "Action": ["s3:PutObject", "s3:GetObject", "s3:DeleteObject"],
      "Resource": ["arn:aws:s3:::example_bucket",
                  "arn:aws:s3:::example_bucket/*"],
      "Condition": {
        "StringNotEquals": {
          "aws:sourceVpc": "vpc-111bbb22"
        }
      }
    }
  ]
}
```

Example Example: Restrict access to a specific IP address range

You can create a policy that restricts access to specific IP address ranges by using the [aws:VpcSourceIp](#) condition key. The following policy denies access to the specified bucket using the specified actions unless the request comes from the specified IP address. Note that this policy blocks access to the specified bucket using the specified actions through the AWS Management Console.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Sid": "Allow-access-to-specific-VPC-CIDR",
    "Effect": "Deny",
    "Principal": "*",
    "Action": ["s3:PutObject", "s3:GetObject", "s3:DeleteObject"],
    "Resource": ["arn:aws:s3:::bucket_name",
                 "arn:aws:s3:::bucket_name/*"],
    "Condition": {
      "NotIpAddress": {
        "aws:VpcSourceIp": "172.31.0.0/16"
      }
    }
  }
]
}

```

Example Example: Restrict access to buckets in a specific AWS account

You can create a policy that restricts access to the S3 buckets in a specific AWS account by using the `s3:ResourceAccount` condition key. The following policy denies access to S3 buckets using the specified actions unless they are owned by the specified AWS account.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow-access-to-bucket-in-specific-account",
      "Effect": "Deny",
      "Principal": "*",
      "Action": ["s3:GetObject", "s3:PutObject", "s3:DeleteObject"],
      "Resource": "arn:aws:s3::*",
      "Condition": {
        "StringNotEquals": {
          "s3:ResourceAccount": "111122223333"
        }
      }
    }
  ]
}

```

Associate route tables

You can change the route tables that are associated with the gateway endpoint. When you associate a route table, we automatically add a route that points traffic destined for the service to

the endpoint network interface. When you disassociate a route table, we automatically remove the endpoint route from the route table.

To associate route tables using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the gateway endpoint.
4. Choose **Actions, Manage route tables**.
5. Select or deselect route tables as needed.
6. Choose **Modify route tables**.

To associate route tables using the command line

- [modify-vpc-endpoint](#) (AWS CLI)
- [Edit-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Edit the VPC endpoint policy

You can edit the endpoint policy for a gateway endpoint, which controls access to Amazon S3 from the VPC through the endpoint. The default policy allows full access. For more information, see [Endpoint policies](#).

To change the endpoint policy using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the gateway endpoint.
4. Choose **Actions, Manage policy**.
5. Choose **Full Access** to allow full access to the service, or choose **Custom** and attach a custom policy.
6. Choose **Save**.

The following are example endpoint policies for accessing Amazon S3.

Example Example: Restrict access to a specific bucket

You can create a policy that restricts access to specific S3 buckets only. This is useful if you have other AWS services in your VPC that use S3 buckets.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow-access-to-specific-bucket",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:ListBucket",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket_name",
        "arn:aws:s3:::bucket_name/*"
      ]
    }
  ]
}
```

Example Example: Restrict access to a specific IAM role

You can create a policy that restricts access to a specific IAM role. You must use `aws:PrincipalArn` to grant access to a principal.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow-access-to-specific-IAM-role",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "ArnEquals": {
          "aws:PrincipalArn": "arn:aws:iam::111122223333:role/role_name"
        }
      }
    }
  ]
}
```

```
    }
  }
]
}
```

Example Example: Restrict access to users in a specific account

You can create a policy that restricts access to a specific account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow-callers-from-specific-account",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalAccount": "111122223333"
        }
      }
    }
  ]
}
```

Delete a gateway endpoint

When you are finished with a gateway endpoint, you can delete it. When you delete a gateway endpoint, we remove the endpoint route from the subnet route tables.

You can't delete a gateway endpoint if private DNS is enabled.

To delete a gateway endpoint using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the gateway endpoint.
4. Choose **Actions, Delete VPC endpoints**.
5. When prompted for confirmation, enter **delete**.

6. Choose **Delete**.

To delete a gateway endpoint using the command line

- [delete-vpc-endpoints](#) (AWS CLI)
- [Remove-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Gateway endpoints for Amazon DynamoDB

You can access Amazon DynamoDB from your VPC using gateway VPC endpoints. After you create the gateway endpoint, you can add it as a target in your route table for traffic destined from your VPC to DynamoDB.

There is no additional charge for using gateway endpoints.

DynamoDB supports both gateway endpoints and interface endpoints. With a gateway endpoint, you can access DynamoDB from your VPC, without requiring an internet gateway or NAT device for your VPC, and with no additional cost. However, gateway endpoints do not allow access from on-premises networks, from peered VPCs in other AWS Regions, or through a transit gateway. For those scenarios, you must use an interface endpoint, which is available for an additional cost. For more information, see [Types of VPC endpoints for DynamoDB](#) in the *Amazon DynamoDB Developer Guide*.

Contents

- [Considerations](#)
- [Create a gateway endpoint](#)
- [Control access using IAM policies](#)
- [Associate route tables](#)
- [Edit the VPC endpoint policy](#)
- [Delete a gateway endpoint](#)

Considerations

- A gateway endpoint is available only in the Region where you created it. Be sure to create your gateway endpoint in the same Region as your DynamoDB tables.

- If you're using the Amazon DNS servers, you must enable both [DNS hostnames and DNS resolution](#) for your VPC. If you're using your own DNS server, ensure that requests to DynamoDB resolve correctly to the IP addresses maintained by AWS.
- The rules for the security groups for your instances that access DynamoDB through a gateway endpoint must allow traffic to and from DynamoDB. You can reference the ID of the [prefix list](#) for DynamoDB in security group rules.
- The network ACL for the subnet for your instances that access DynamoDB through a gateway endpoint must allow traffic to and from DynamoDB. You can't reference prefix lists in network ACL rules, but you can get the IP address range for DynamoDB from the [prefix list](#) for DynamoDB.
- If you use AWS CloudTrail to log DynamoDB operations, the log files contain the private IP addresses of the EC2 instances in the service consumer VPC and the ID of the gateway endpoint for any requests performed through the endpoint.
- Gateway endpoints support only IPv4 traffic.
- The source IPv4 addresses from instances in your affected subnets change from public IPv4 addresses to private IPv4 addresses from your VPC. An endpoint switches network routes and disconnects open TCP connections. The previous connections that used public IPv4 addresses are not resumed. We recommend that you do not have any critical tasks running when you create or modify a gateway endpoint. Alternatively, test to ensure that your software can automatically reconnect to DynamoDB if a connection breaks.
- Endpoint connections cannot be extended out of a VPC. Resources on the other side of a VPN connection, VPC peering connection, transit gateway, or AWS Direct Connect connection in your VPC cannot use a gateway endpoint to communicate with DynamoDB.
- Your account has a default quota of 20 gateway endpoints per Region, which is adjustable. There is also a limit of 255 gateway endpoints per VPC.

Create a gateway endpoint

Use the following procedure to create a gateway endpoint that connects to DynamoDB.

To create a gateway endpoint using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create endpoint**.

4. For **Service category**, choose **AWS services**.
5. For **Services**, add the filter **Type: Gateway** and select **com.amazonaws.*region*.dynamodb**.
6. For **VPC**, select the VPC in which to create the endpoint.
7. For **Route tables**, select the route tables to be used by the endpoint. We automatically add a route that points traffic destined for the service to the endpoint network interface.
8. For **Policy**, select **Full access** to allow all operations by all principals on all resources over the VPC endpoint. Otherwise, select **Custom** to attach a VPC endpoint policy that controls the permissions that principals have to perform actions on resources over the VPC endpoint.
9. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
10. Choose **Create endpoint**.

To create a gateway endpoint using the command line

- [create-vpc-endpoint](#) (AWS CLI)
- [New-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Control access using IAM policies

You can create IAM policies to control which IAM principals can access DynamoDB tables using a specific VPC endpoint.

Example Example: Restrict access to a specific endpoint

You can create a policy that restricts access to a specific VPC endpoint by using the [aws:sourceVpce](#) condition key. The following policy denies access to DynamoDB tables in the account unless the specified VPC endpoint is used. This example assumes that there is also a policy statement that allows the access required for your use cases.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow-access-from-specific-endpoint",
      "Effect": "Deny",
      "Action": "dynamodb:*",
      "Resource": "arn:aws:dynamodb:region:account-id:table/*",
      "Condition": {
```

```

        "StringNotEquals" : {
            "aws:sourceVpce": "vpce-11aa22bb"
        }
    }
}
]
}

```

Example Example: Allow access from a specific IAM role

You can create a policy that allows access using a specific IAM role. The following policy grants access to the specified IAM role.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow-access-from-specific-IAM-role",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "ArnEquals": {
          "aws:PrincipalArn": "arn:aws:iam::111122223333:role/role_name"
        }
      }
    }
  ]
}

```

Example Example: Allows access from a specific account

You can create a policy that allows access from a specific account only. The following policy grants access to users in the specified account.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow-access-from-account",
      "Effect": "Allow",

```

```
    "Principal": "*",
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:PrincipalAccount": "111122223333"
      }
    }
  }
]
```

Associate route tables

You can change the route tables that are associated with the gateway endpoint. When you associate a route table, we automatically add a route that points traffic destined for the service to the endpoint network interface. When you disassociate a route table, we automatically remove the endpoint route from the route table.

To associate route tables using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the gateway endpoint.
4. Choose **Actions, Manage route tables**.
5. Select or deselect route tables as needed.
6. Choose **Modify route tables**.

To associate route tables using the command line

- [modify-vpc-endpoint](#) (AWS CLI)
- [Edit-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Edit the VPC endpoint policy

You can edit the endpoint policy for a gateway endpoint, which controls access to DynamoDB from the VPC through the endpoint. The default policy allows full access. For more information, see [Endpoint policies](#).

To change the endpoint policy using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the gateway endpoint.
4. Choose **Actions, Manage policy**.
5. Choose **Full Access** to allow full access to the service, or choose **Custom** and attach a custom policy.
6. Choose **Save**.

To modify a gateway endpoint using the command line

- [modify-vpc-endpoint](#) (AWS CLI)
- [Edit-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

The following are example endpoint policies for accessing DynamoDB.

Example Example: Allow read-only access

You can create a policy that restricts access to read-only access. The following policy grants permission to list and describe DynamoDB tables.

```
{
  "Statement": [
    {
      "Sid": "ReadOnlyAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:ListTables"
      ],
      "Resource": "*"
    }
  ]
}
```

Example Example: Restrict access to a specific table

You can create a policy that restricts access to a specific DynamoDB table. The following policy allows access to the specified DynamoDB table.

```
{
  "Statement": [
    {
      "Sid": "Allow-access-to-specific-table",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "dynamodb:Batch*",
        "dynamodb:Delete*",
        "dynamodb:DescribeTable",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:Update*"
      ],
      "Resource": "arn:aws:dynamodb:region:123456789012:table/table_name"
    }
  ]
}
```

Delete a gateway endpoint

When you are finished with a gateway endpoint, you can delete it. When you delete a gateway endpoint, we remove the endpoint route from the subnet route tables.

To delete a gateway endpoint using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the gateway endpoint.
4. Choose **Actions, Delete VPC endpoints**.
5. When prompted for confirmation, enter **delete**.
6. Choose **Delete**.

To delete a gateway endpoint using the command line

- [delete-vpc-endpoints](#) (AWS CLI)
- [Remove-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Access SaaS products through AWS PrivateLink

Using AWS PrivateLink, you can access SaaS products privately, as if they were running in your own VPC.

Contents

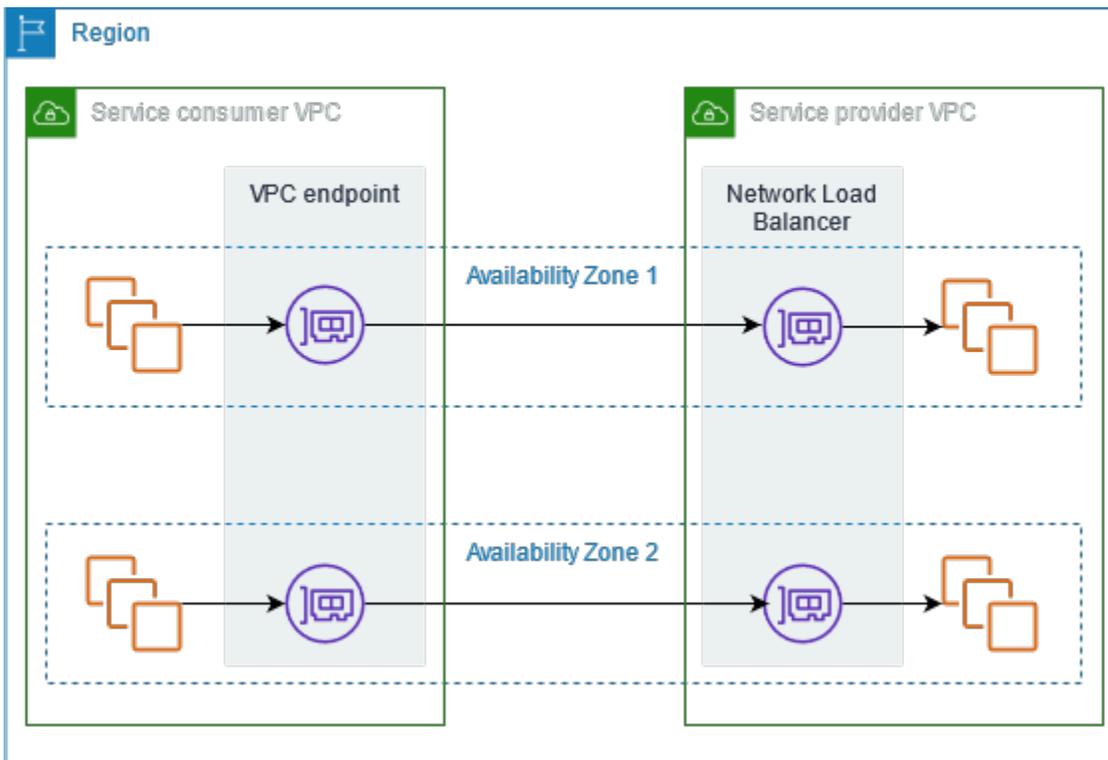
- [Overview](#)
- [Create an interface endpoint](#)

Overview

You can discover, purchase, and provision SaaS products powered by AWS PrivateLink through AWS Marketplace. For more information, see [AWS Marketplace: - PrivateLink](#).

You can also find SaaS products powered by AWS PrivateLink from AWS Partners. For more information see [AWS PrivateLink Partners](#).

The following diagram shows how you use VPC endpoints to connect to SaaS products. The service provider creates an endpoint service and grants their customers access to the endpoint service. As the service consumer, you create an interface VPC endpoint, which establishes connections between one or more subnets in your VPC and the endpoint service.



Create an interface endpoint

Use the following procedure to create an interface VPC endpoint that connects to the SaaS product.

Requirement

Subscribe to the service.

To create an interface endpoint to a partner service

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create endpoint**.
4. If you purchased the service from AWS Marketplace, do the following:
 - a. For **Service category**, choose **AWS Marketplace services**.
 - b. Enter the name of the service.
5. If you subscribed to a service with the AWS Service Ready designation, do the following:

- a. For **Service category**, choose **PrivateLink Ready partner services**.
- b. Enter the name of the service and choose **Verify service**.
6. For **VPC**, select the VPC from which you'll access the product.
7. For **Subnets**, select one subnet per Availability Zone from which you'll access the product.
8. For **Security group**, select the security groups to associate with the endpoint network interfaces. The security group rules must allow traffic between the resources in the VPC and the endpoint network interfaces.
9. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
10. Choose **Create endpoint**.

To configure an interface endpoint

For information about configuring your interface endpoint, see [the section called "Configure an interface endpoint"](#).

Access virtual appliances through AWS PrivateLink

You can use a Gateway Load Balancer to distribute traffic to a fleet of network virtual appliances. The appliances can be used for security inspection, compliance, policy controls, and other networking services. You specify the Gateway Load Balancer when you create a VPC endpoint service. Other AWS principals access the endpoint service by creating a Gateway Load Balancer endpoint.

Pricing

You are billed for each hour that your Gateway Load Balancer endpoint is provisioned in each Availability Zone. You are also billed per GB of data processed. For more information, see [AWS PrivateLink Pricing](#).

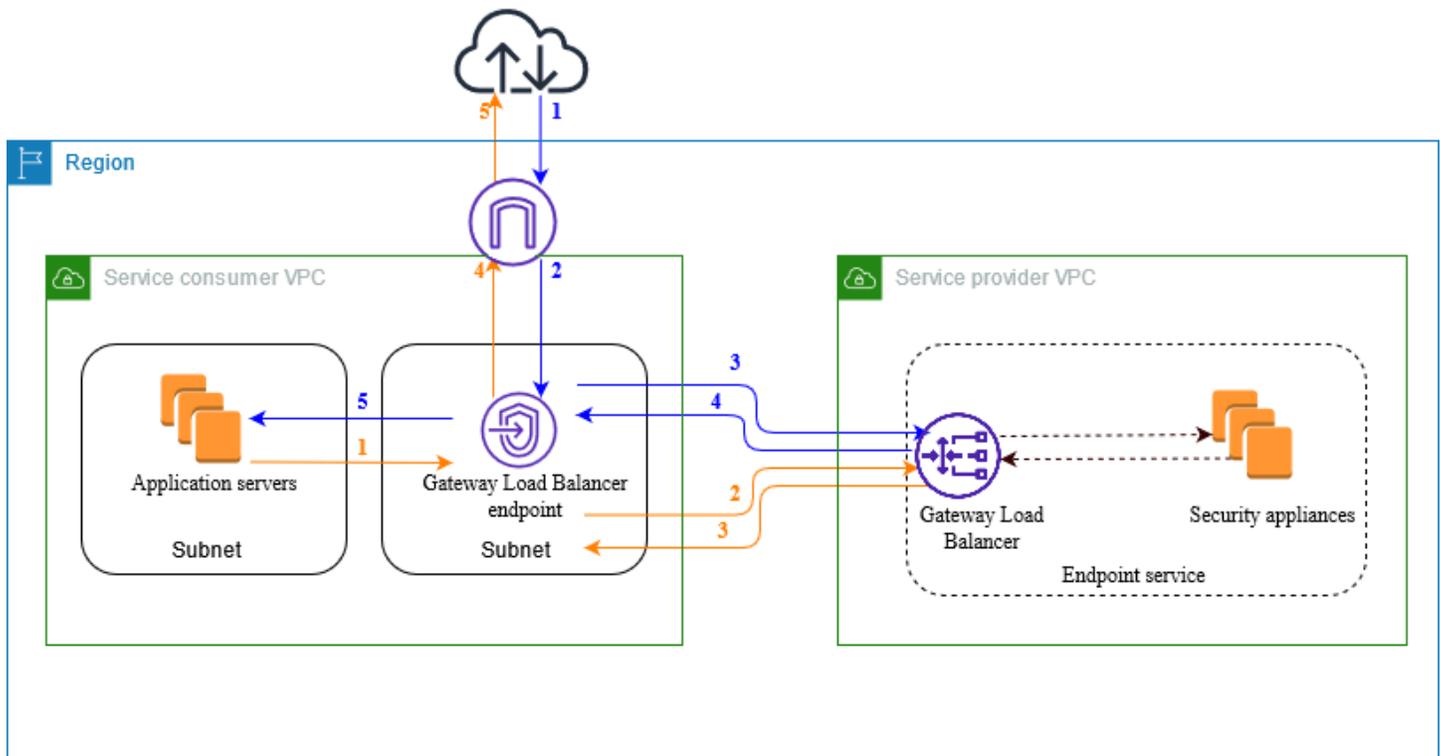
Contents

- [Overview](#)
- [IP address types](#)
- [Routing](#)
- [Create an inspection system as a Gateway Load Balancer endpoint service](#)
- [Access an inspection system using a Gateway Load Balancer endpoint](#)

For more information, see [Gateway Load Balancers](#).

Overview

The following diagram shows how application servers access security appliances through AWS PrivateLink. The application servers run in a subnet of the service consumer VPC. You create a Gateway Load Balancer endpoint in another subnet of the same VPC. All traffic entering the service consumer VPC through the internet gateway is first routed to the Gateway Load Balancer endpoint for inspection and then routed to the destination subnet. Similarly, all traffic leaving the application servers is routed to the Gateway Load Balancer endpoint for inspection before it is routed back through the internet gateway.



Traffic from the internet to the application servers (blue arrows):

1. Traffic enters the service consumer VPC through the internet gateway.
2. Traffic is sent to the Gateway Load Balancer endpoint, based on route table configuration.
3. Traffic is sent to the Gateway Load Balancer for inspection through the security appliance.
4. Traffic is sent back to the Gateway Load Balancer endpoint after inspection.
5. Traffic is sent to the application servers, based on route table configuration.

Traffic from the application servers to the internet (orange arrows):

1. Traffic is sent to the Gateway Load Balancer endpoint, based on route table configuration.
2. Traffic is sent to the Gateway Load Balancer for inspection through the security appliance.
3. Traffic is sent back to the Gateway Load Balancer endpoint after inspection.
4. Traffic is sent to the internet gateway based on the route table configuration.
5. Traffic is routed back to the internet.

IP address types

Service providers can make their service endpoints available to service consumers over IPv4, IPv6, or both IPv4 and IPv6, even if their security appliances support only IPv4. If you enable dualstack support, existing consumers can continue to use IPv4 to access your service and new consumers can choose to use IPv6 to access your service.

If a Gateway Load Balancer endpoint supports IPv4, the endpoint network interfaces have IPv4 addresses. If a Gateway Load Balancer endpoint supports IPv6, the endpoint network interfaces have IPv6 addresses. The IPv6 address for an endpoint network interface is unreachable from the internet. If you describe an endpoint network interface with an IPv6 address, notice that `denyAllIgwTraffic` is enabled.

Requirements to enable IPv6 for an endpoint service

- The VPC and subnets for the endpoint service must have associated IPv6 CIDR blocks.
- The Gateway Load Balancer for the endpoint service must use the dualstack IP address type. The security appliances do not need to support IPv6 traffic.

Requirements to enable IPv6 for a Gateway Load Balancer endpoint

- The endpoint service must have an IP address type that includes IPv6 support.
- The IP address type of a Gateway Load Balancer endpoint must be compatible with the subnet for the Gateway Load Balancer endpoint, as described here:
 - **IPv4** – Assign IPv4 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have IPv4 address ranges.
 - **IPv6** – Assign IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets are IPv6 only subnets.
 - **Dualstack** – Assign both IPv4 and IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have both IPv4 and IPv6 address ranges.
- The route tables for the subnets in the service consumer VPC must route IPv6 traffic and the network ACLs for these subnets must allow IPv6 traffic.

Routing

To route traffic to the endpoint service, specify the Gateway Load Balancer endpoint as a target in your route tables, using its ID. For the diagram above, add routes to the route tables as follows. Note that IPv6 routes are included for a dualstack configuration.

Route table for the internet gateway

This route table must have a route that sends traffic destined for the application servers to the Gateway Load Balancer endpoint.

Destination	Target
<i>VPC IPv4 CIDR</i>	Local
<i>VPC IPv6 CIDR</i>	Local
<i>Application subnet IPv4 CIDR</i>	<i>vpc-endpoint-id</i>
<i>Application subnet IPv6 CIDR</i>	<i>vpc-endpoint-id</i>

Route table for the subnet with the application servers

This route table must have a route that sends all traffic from the application servers to the Gateway Load Balancer endpoint.

Destination	Target
<i>VPC IPv4 CIDR</i>	Local
<i>VPC IPv6 CIDR</i>	Local
0.0.0.0/0	<i>vpc-endpoint-id</i>
::/0	<i>vpc-endpoint-id</i>

Route table for the subnet with the Gateway Load Balancer endpoint

This route table must send traffic that is returned from inspection to its final destination. For traffic that originated from the internet, the local route sends the traffic to the application servers. For traffic that originated from the application servers, add a route that sends all traffic to the internet gateway.

Destination	Target
<i>VPC IPv4 CIDR</i>	Local
<i>VPC IPv6 CIDR</i>	Local
0.0.0.0/0	<i>internet-gateway-id</i>
::/0	<i>internet-gateway-id</i>

Create an inspection system as a Gateway Load Balancer endpoint service

You can create your own service powered by AWS PrivateLink, known as an *endpoint service*. You are the service provider, and the AWS principals that create connections to your service are the service consumers.

Endpoint services require either a Network Load Balancer or a Gateway Load Balancer. In this case, you'll create an endpoint service using a Gateway Load Balancer. For more information about creating an endpoint service using a Network Load Balancer, see [Create an endpoint service](#).

Contents

- [Considerations](#)
- [Prerequisites](#)
- [Create the endpoint service](#)
- [Make your endpoint service available](#)

Considerations

- An endpoint service is available in the Region where you created it.

- When service consumers retrieve information about an endpoint service, they can see only the Availability Zones that they have in common with the service provider. When the service provider and service consumer are in different accounts, an Availability Zone name, such as us-east-1a, might be mapped to a different physical Availability Zone in each AWS account. You can use AZ IDs to consistently identify the Availability Zones for your service. For more information, see [AZ IDs](#) in the *Amazon EC2 User Guide for Linux Instances*.
- There are quotas on your AWS PrivateLink resources. For more information, see [AWS PrivateLink quotas](#).

Prerequisites

- Create a service provider VPC with at least two subnets in the Availability Zone in which the service should be available. One subnet is for the security appliance instances and the other is for the Gateway Load Balancer.
- Create a Gateway Load Balancer in your service provider VPC. If you plan to enable IPv6 support on your endpoint service, you must enable dualstack support on your Gateway Load Balancer. For more information, see [Getting started with Gateway Load Balancers](#).
- Launch security appliances in the service provider VPC and register them with a load balancer target group.

Create the endpoint service

Use the following procedure to create an endpoint service using a Gateway Load Balancer.

To create an endpoint service using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Choose **Create endpoint service**.
4. For **Load balancer type**, choose **Gateway**.
5. For **Available load balancers**, select your Gateway Load Balancer.
6. For **Require acceptance for endpoint**, select **Acceptance required** to require that connection requests to your endpoint service are accepted manually. Otherwise, they are accepted automatically.
7. For **Supported IP address types**, do one of the following:

- Select **IPv4** – Enable the endpoint service to accept IPv4 requests.
 - Select **IPv6** – Enable the endpoint service to accept IPv6 requests.
 - Select **IPv4 and IPv6** – Enable the endpoint service to accept both IPv4 and IPv6 requests.
8. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
 9. Choose **Create**.

To create an endpoint service using the command line

- [create-vpc-endpoint-service-configuration](#) (AWS CLI)
- [New-EC2VpcEndpointServiceConfiguration](#) (Tools for Windows PowerShell)

Make your endpoint service available

Service providers must do the following to make their services available to service consumers.

- Add permissions that allow each service consumer to connect to your endpoint service. For more information, see [the section called “Manage permissions”](#).
- Provide the service consumer with the name of your service and the supported Availability Zones so that they can create an interface endpoint to connect to your service. For more information, see the procedure below.
- Accept the endpoint connection request from the service consumer. For more information see [the section called “Accept or reject connection requests”](#).

AWS principals can connect to your endpoint service privately by creating a Gateway Load Balancer endpoint. For more information, see [Create a Gateway Load Balancer endpoint](#).

Access an inspection system using a Gateway Load Balancer endpoint

You can create a Gateway Load Balancer endpoint to connect to [endpoint services](#) powered by AWS PrivateLink.

For each subnet that you specify from your VPC, we create an endpoint network interface in the subnet and assign it a private IP address from the subnet address range. An endpoint network

interface is a requester-managed network interface; you can view it in your AWS account, but you can't manage it yourself.

You are billed for hourly usage and data processing charges. For more information, see [Gateway Load Balancer endpoint pricing](#).

Contents

- [Considerations](#)
- [Prerequisites](#)
- [Create the endpoint](#)
- [Configure routing](#)
- [Manage tags](#)
- [Delete a Gateway Load Balancer endpoint](#)

Considerations

- You can choose only one Availability Zone in the service consumer VPC. You can't change this subnet later on. To use a Gateway Load Balancer endpoint in a different subnet, you must create a new Gateway Load Balancer endpoint.
- You can create a single Gateway Load Balancer endpoint per Availability Zone per service, and you must select the Availability Zone that the Gateway Load Balancer supports. When the service provider and service consumer are in different accounts, an Availability Zone name, such as `us-east-1a`, might be mapped to a different physical Availability Zone in each AWS account. You can use AZ IDs to consistently identify the Availability Zones for your service. For more information, see [AZ IDs](#) in the *Amazon EC2 User Guide for Linux Instances*.
- Before you can use the endpoint service the service provider must accept the connection requests. The service can't initiate requests to resources in your VPC through the VPC endpoint. The endpoint only returns responses to traffic that was initiated by resources in your VPC.
- Each Gateway Load Balancer endpoint can support a bandwidth of up to 10 Gbps per Availability Zone and automatically scales up to 100 Gbps.
- If an endpoint service is associated with multiple Gateway Load Balancers, a Gateway Load Balancer endpoint establishes a connection with only one load balancer per Availability Zone.
- To keep traffic within the same Availability Zone, we recommend that you create a Gateway Load Balancer endpoint in each Availability Zone to which you'll send traffic.

- Network Load Balancer client IP preservation is not supported when traffic is routed through a Gateway Load Balancer endpoint, even if the target is in the same VPC as the Network Load Balancer.
- There are quotas on your AWS PrivateLink resources. For more information, see [AWS PrivateLink quotas](#).

Prerequisites

- Create a service consumer VPC with at least two subnets in the Availability Zone from which you'll access the service. One subnet is for the application servers and the other is for the Gateway Load Balancer endpoint.
- To verify which Availability Zones are supported by the endpoint service, describe the endpoint service using the console or the [describe-vpc-endpoint-services](#) command.
- If your resources are in a subnet with a network ACL, verify that the network ACL allows traffic between the endpoint network interfaces and the resources in the VPC.

Create the endpoint

Use the following procedure to create a Gateway Load Balancer endpoint that connects to the endpoint service for the inspection system.

To create a Gateway Load Balancer endpoint using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create endpoint**.
4. For **Service category**, choose **Other endpoint services**.
5. For **Service name**, enter the name of the service, and then choose **Verify service**.
6. For **VPC**, select the VPC in which to create the endpoint.
7. For **Subnets**, select the subnet in which to create the endpoint.
8. For **IP address type**, choose from the following options:
 - **IPv4** – Assign IPv4 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have IPv4 address ranges.

- **IPv6** – Assign IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets are IPv6 only subnets.
 - **Dualstack** – Assign both IPv4 and IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have both IPv4 and IPv6 address ranges.
9. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
 10. Choose **Create endpoint**. The initial status is pending acceptance.

To create a Gateway Load Balancer endpoint using the command line

- [create-vpc-endpoint](#) (AWS CLI)
- [New-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Configure routing

Use the following procedure to configure route tables for the service consumer VPC. This enables the security appliances to perform security inspection for inbound traffic that's destined for the application servers. For more information, see [the section called "Routing"](#).

To configure routing using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Route Tables**.
3. Select the route table for the internet gateway and do the following:
 - a. Choose **Actions, Edit routes**.
 - b. If you support IPv4, choose **Add route**. For **Destination**, enter the IPv4 CIDR block of the subnet for the application servers. For **Target**, select the VPC endpoint.
 - c. If you support IPv6, choose **Add route**. For **Destination**, enter the IPv6 CIDR block of the subnet for the application servers. For **Target**, select the VPC endpoint.
 - d. Choose **Save changes**.
4. Select the route table for the subnet with the application servers and do the following:
 - a. Choose **Actions, Edit routes**.
 - b. If you support IPv4, choose **Add route**. For **Destination**, enter **0.0.0.0/0**. For **Target**, select the VPC endpoint.

- c. If you support IPv6, choose **Add route**. For **Destination**, enter **::/0**. For **Target**, select the VPC endpoint.
 - d. Choose **Save changes**.
5. Select the route table for the subnet with the Gateway Load Balancer endpoint, and do the following:
 - a. Choose **Actions, Edit routes**.
 - b. If you support IPv4, choose **Add route**. For **Destination**, enter **0.0.0.0/0**. For **Target**, select the internet gateway.
 - c. If you support IPv6, choose **Add route**. For **Destination**, enter **::/0**. For **Target**, select the internet gateway.
 - d. Choose **Save changes**.

To configure routing using the command line

- [create-route](#) (AWS CLI)
- [New-EC2Route](#) (Tools for Windows PowerShell)

Manage tags

You can tag your Gateway Load Balancer endpoint to help you identify it or categorize it according to your organization's needs.

To manage tags using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the interface endpoint.
4. Choose **Actions, Manage tags**.
5. For each tag to add choose **Add new tag** and enter the tag key and tag value.
6. To remove a tag, choose **Remove** to the right of the tag key and value.
7. Choose **Save**.

To manage tags using the command line

- [create-tags](#) and [delete-tags](#) (AWS CLI)
- [New-EC2Tag](#) and [Remove-EC2Tag](#) (Tools for Windows PowerShell)

Delete a Gateway Load Balancer endpoint

When you are finished with an endpoint, you can delete it. Deleting a Gateway Load Balancer endpoint also deletes the endpoint network interfaces. You can't delete a Gateway Load Balancer endpoint if there are routes in your route tables that point to the endpoint.

To delete a Gateway Load Balancer endpoint

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints** and select your endpoint.
3. Choose **Actions, Delete Endpoint**.
4. In the confirmation screen, choose **Yes, Delete**.

To delete a Gateway Load Balancer endpoint

- [delete-vpc-endpoints](#) (AWS CLI)
- [Remove-EC2VpcEndpoint](#) (AWS Tools for Windows PowerShell)

Share your services through AWS PrivateLink

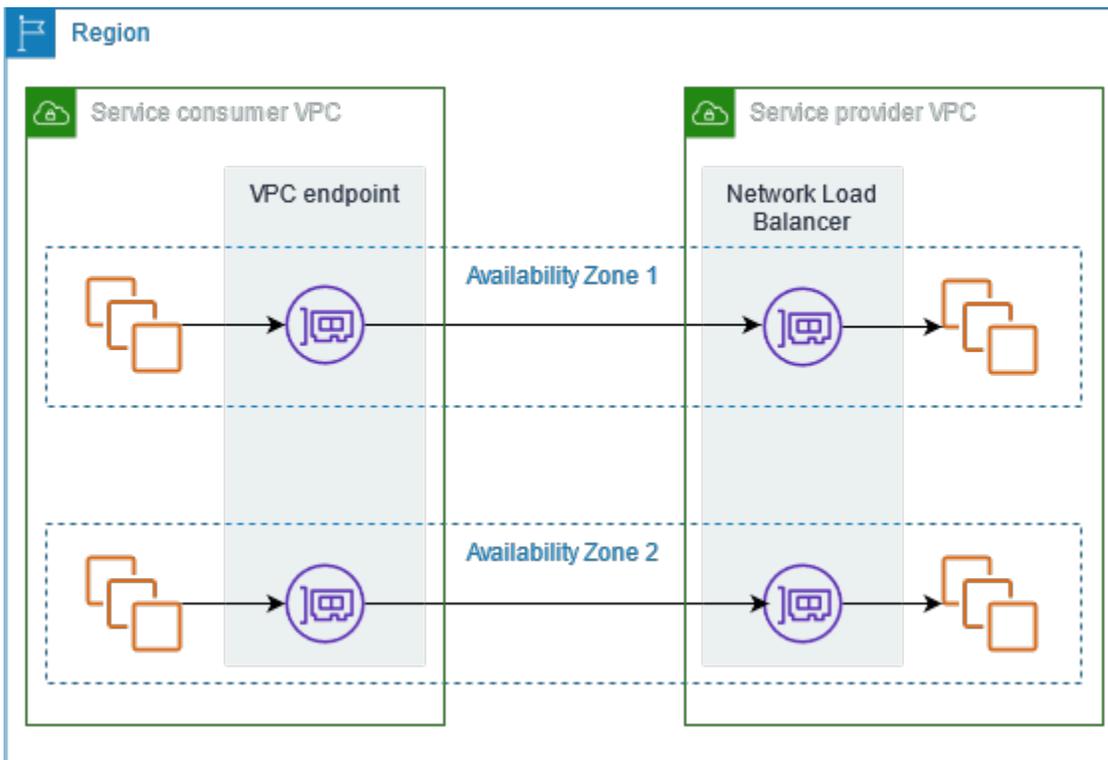
You can host your own AWS PrivateLink powered service, known as an *endpoint service*, and share it with other AWS customers.

Contents

- [Overview](#)
- [DNS hostnames](#)
- [Private DNS](#)
- [IP address types](#)
- [Create a service powered by AWS PrivateLink](#)
- [Configure an endpoint service](#)
- [Manage DNS names for VPC endpoint services](#)
- [Receive alerts for endpoint service events](#)
- [Delete an endpoint service](#)

Overview

The following diagram shows how you share your service that's hosted in AWS with other AWS customers, and how those customers connect to your service. As the service provider, you create a Network Load Balancer in your VPC as the service front end. You then select this load balancer when you create the VPC endpoint service configuration. You grant permission to specific AWS principals so that they can connect to your service. As a service consumer, the customer creates an interface VPC endpoint, which establishes connections between the subnets that they select from their VPC and your endpoint service. The load balancer receives requests from the service consumer and routes them to the targets hosting your service.



For low latency and high availability, we recommend that you make your service available in at least two Availability Zones.

DNS hostnames

When a service provider creates a VPC endpoint service, AWS generates an endpoint-specific DNS hostname for the service. These names have the following syntax:

```
endpoint_service_id.region.vpce.amazonaws.com
```

The following is an example of a DNS hostname for a VPC endpoint service in the us-east-2 Region:

```
vpce-svc-071afff70666e61e0.us-east-2.vpce.amazonaws.com
```

When a service consumer creates an interface VPC endpoint, we create Regional and zonal DNS names that the service consumer can use to communicate with the endpoint service. Regional names have the following syntax:

```
endpoint_id.endpoint_service_id.region.vpce.amazonaws.com
```

Zonal names have the following syntax:

```
endpoint_id-zone.endpoint_service_id.region.vpce.amazonaws.com
```

Private DNS

A service provider can also associate a private DNS name for their endpoint service, so that service consumers can continue to access the service using its existing DNS name. If a service provider associates a private DNS name with their endpoint service, then service consumers can enable private DNS names for their interface endpoints. If a service provider doesn't enable private DNS, then service consumers might need to update their applications to use the public DNS name of the VPC endpoint service. For more information, see [Manage DNS names](#).

IP address types

Service providers can make their service endpoints available to service consumers over IPv4, IPv6, or both IPv4 and IPv6, even if their backend servers support only IPv4. If you enable dualstack support, existing consumers can continue to use IPv4 to access your service and new consumers can choose to use IPv6 to access your service.

If an interface VPC endpoint supports IPv4, the endpoint network interfaces have IPv4 addresses. If an interface VPC endpoint supports IPv6, the endpoint network interfaces have IPv6 addresses. The IPv6 address for an endpoint network interface is unreachable from the internet. If you describe an endpoint network interface with an IPv6 address, notice that `denyAllIgwTraffic` is enabled.

Requirements to enable IPv6 for an endpoint service

- The VPC and subnets for the endpoint service must have associated IPv6 CIDR blocks.
- All Network Load Balancers for the endpoint service must use the dualstack IP address type. The targets do not need to support IPv6 traffic. If the service processes source IP addresses from the proxy protocol version 2 header, it must process IPv6 addresses.

Requirements to enable IPv6 for an interface endpoint

- The endpoint service must support IPv6 requests.

- The IP address type of an interface endpoint must be compatible with the subnets for the interface endpoint, as described here:
 - **IPv4** – Assign IPv4 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have IPv4 address ranges.
 - **IPv6** – Assign IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets are IPv6 only subnets.
 - **Dualstack** – Assign both IPv4 and IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have both IPv4 and IPv6 address ranges.

DNS record IP address type for an interface endpoint

The DNS record IP address type that an interface endpoint supports determines the DNS records that we create. The DNS record IP address type of an interface endpoint must be compatible with the IP address type of the interface endpoint, as described here:

- **IPv4** – Create A records for the private, Regional, and zonal DNS names. The IP address type must be **IPv4** or **Dualstack**.
- **IPv6** – Create AAAA records for the private, Regional, and zonal DNS names. The IP address type must be **IPv6** or **Dualstack**.
- **Dualstack** – Create A and AAAA records for the private, Regional, and zonal DNS names. The IP address type must be **Dualstack**.

Create a service powered by AWS PrivateLink

You can create your own service powered by AWS PrivateLink, known as an *endpoint service*. You are the service provider, and the AWS principals that create connections to your service are the service consumers.

Endpoint services require either a Network Load Balancer or a Gateway Load Balancer. The load balancer receives requests from service consumers and routes them to your service. In this case, you'll create an endpoint service using a Network Load Balancer. For more information about creating an endpoint service using a Gateway Load Balancer, see [Access virtual appliances](#).

Contents

- [Considerations](#)
- [Prerequisites](#)

- [Create an endpoint service](#)
- [Make your endpoint service available to service consumers](#)

Considerations

- An endpoint service is available in the Region where you created it. You can access the endpoint service from other Regions using VPC peering.
- An endpoint service supports traffic only over TCP.
- When service consumers retrieve information about an endpoint service, they can see only the Availability Zones that they have in common with the service provider. When the service provider and service consumer are in different accounts, an Availability Zone name, such as us-east-1a, might be mapped to a different physical Availability Zone in each AWS account. You can use AZ IDs to consistently identify the Availability Zones for your service. For more information, see [AZ IDs](#) in the *Amazon EC2 User Guide for Linux Instances*.
- When service consumers send traffic to a service through an interface endpoint, the source IP addresses provided to the application are the private IP addresses of the load balancer nodes, not the IP addresses of the service consumers. If you enable proxy protocol on the load balancer, you can obtain the addresses of the service consumers and the IDs of the interface endpoints from the proxy protocol header. For more information, see [Proxy protocol](#) in the *User Guide for Network Load Balancers*.
- If an endpoint service is associated with multiple Network Load Balancers, each endpoint network interface is associated with one load balancer. When the first connection from an endpoint network interface is initiated, we select one of the Network Load Balancers in the same Availability Zone as the endpoint network interface at random. All subsequent connection requests from this endpoint network interface use the selected load balancer. We recommend that you use the same listener and target group configuration for all load balancers for an endpoint service, so that consumers can use the endpoint service successfully no matter which load balancer is chosen.
- There are quotas on your AWS PrivateLink resources. For more information, see [AWS PrivateLink quotas](#).

Prerequisites

- Create a VPC for your endpoint service with at least one subnet in each Availability Zone in which the service should be available.

- To enable service consumers to create IPv6 interface VPC endpoints for your endpoint service, the VPC and subnets must have associated IPv6 CIDR blocks.
- Create a Network Load Balancer in your VPC. Select one subnet per Availability Zone in which the service should be available to service consumers. For low latency and fault tolerance, we recommend that you make your service available in at least two Availability Zones in the Region.
- To enable your endpoint service to accept IPv6 requests, its Network Load Balancers must use the dualstack IP address type. The targets do not need to support IPv6 traffic. For more information, see [IP address type](#) in the *User Guide for Network Load Balancers*.

If you process source IP addresses from the proxy protocol version 2 header, verify that you can process IPv6 addresses.

- Launch instances in each Availability Zone in which the service should be available and register them with a load balancer target group. If you do not launch instances in all enabled Availability Zones, you can enable cross-zone load balancing to support service consumers that use zonal DNS hostnames to access the service. Regional data transfer charges apply when you enable cross-zone load balancing. For more information, see [Cross-zone load balancing](#) in the *User Guide for Network Load Balancers*

Create an endpoint service

Use the following procedure to create an endpoint service using a Network Load Balancer.

To create an endpoint service using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Choose **Create endpoint service**.
4. For **Load balancer type**, choose **Network**.
5. For **Available load balancers**, select the Network Load Balancers to associate with the endpoint service.
6. For **Require acceptance for endpoint**, select **Acceptance required** to require that connection requests to your endpoint service are accepted manually. Otherwise, these requests are accepted automatically.
7. For **Enable private DNS name**, select **Associate a private DNS name with the service** to associate a private DNS name that service consumers can use to access your service, and then

enter the private DNS name. Otherwise, service consumers can use the endpoint-specific DNS name provided by AWS. Before service consumers can use the private DNS name, the service provider must verify that they own the domain. For more information, see [Manage DNS names](#).

8. For **Supported IP address types**, do one of the following:
 - Select **IPv4** – Enable the endpoint service to accept IPv4 requests.
 - Select **IPv6** – Enable the endpoint service to accept IPv6 requests.
 - Select **IPv4 and IPv6** – Enable the endpoint service to accept both IPv4 and IPv6 requests.
9. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
10. Choose **Create**.

To create an endpoint service using the command line

- [create-vpc-endpoint-service-configuration](#) (AWS CLI)
- [New-EC2VpcEndpointServiceConfiguration](#) (Tools for Windows PowerShell)

Make your endpoint service available to service consumers

AWS principals can connect to your endpoint service privately by creating an interface VPC endpoint. Service providers must do the following to make their services available to service consumers.

- Add permissions that allow each service consumer to connect to your endpoint service. For more information, see [the section called “Manage permissions”](#).
- Provide the service consumer with the name of your service and the supported Availability Zones so that they can create an interface endpoint to connect to your service. For more information, see the following procedure.
- Accept the endpoint connection request from the service consumer. For more information, see [the section called “Accept or reject connection requests”](#).

Connect to an endpoint service as the service consumer

A service consumer uses the following procedure to create an interface endpoint to connect to your endpoint service.

To create an interface endpoint using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create endpoint**.
4. For **Service category**, choose **Other endpoint services**.
5. For **Service name**, enter the name of the service (for example, `com.amazonaws.vpce.us-east-1.vpce-svc-0e123abc123198abc`), and choose **Verify service**.
6. For **VPC**, select a VPC in which to create the endpoint.
7. For **Subnets**, select the subnets (Availability Zones) from which you'll access the endpoint service.
8. For **IP address type**, choose from the following options:
 - **IPv4** – Assign IPv4 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have IPv4 address ranges and the endpoint service accepts IPv4 requests.
 - **IPv6** – Assign IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets are IPv6 only subnets and the endpoint service accepts IPv6 requests.
 - **Dualstack** – Assign both IPv4 and IPv6 addresses to your endpoint network interfaces. This option is supported only if all selected subnets have both IPv4 and IPv6 address ranges and the endpoint service accepts both IPv4 and IPv6 requests.
9. For **DNS record IP type**, choose from the following options:
 - **IPv4** – Create A records for the private, Regional, and zonal DNS names. The IP address type must be **IPv4** or **Dualstack**.
 - **IPv6** – Create AAAA records for the private, Regional, and zonal DNS names. The IP address type must be **IPv6** or **Dualstack**.
 - **Dualstack** – Create A and AAAA records for the private, Regional, and zonal DNS names. The IP address type must be **Dualstack**.
 - **Service defined** – Create A records for the private, Regional, and zonal DNS names and AAAA records for the Regional and zonal DNS names. The IP address type must be **Dualstack**.
10. For **Security group**, select the security groups to associate with the endpoint network interfaces.

11. Choose **Create endpoint**.

To create an interface endpoint using the command line

- [create-vpc-endpoint](#) (AWS CLI)
- [New-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

Configure an endpoint service

After you create an endpoint service, you can update its configuration.

Tasks

- [Manage permissions](#)
- [Accept or reject connection requests](#)
- [Change the load balancer association](#)
- [Associate a private DNS name](#)
- [Modify the supported IP address types](#)
- [Manage tags](#)

Manage permissions

The combination of permissions and acceptance settings help you control which service consumers (AWS principals) can access your endpoint service. For example, you can grant permissions to specific principals that you trust and automatically accept all connection requests, or you can grant permissions to a wider group of principals and manually accept specific connection requests that you trust.

By default, your endpoint service is not available to service consumers. You must add permissions that allow specific AWS principals to create an interface VPC endpoint to connect to your endpoint service. To add permissions for an AWS principal, you need its Amazon Resource Name (ARN). The following list includes example ARNs for supported AWS principals.

ARNs for AWS principals

AWS account (includes all principals in the account)

```
arn:aws:iam::account_id:root
```

Role

arn:aws:iam::*account_id*:role/*role_name*

User

arn:aws:iam::*account_id*:user/*user_name*

All principals in all AWS accounts

*

Consideration

If you grant everyone permission to access the endpoint service and configure the endpoint service to accept all requests, your load balancer will be public even if it has no public IP address.

To manage permissions for your endpoint service using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select the endpoint service and choose the **Allow principals** tab.
4. To add permissions, choose **Allow principals**. For **Principals to add**, enter the ARN of the principal. To add another principal, choose **Add principal**. When you are finished adding principals, choose **Allow principals**.
5. To remove permissions, select the principal and choose **Actions, Delete**. When prompted for confirmation, enter **delete** and then choose **Delete**.

To add permissions for your endpoint service using the command line

- [modify-vpc-endpoint-service-permissions](#) (AWS CLI)
- [Edit-EC2EndpointServicePermission](#) (Tools for Windows PowerShell)

Accept or reject connection requests

The combination of permissions and acceptance settings help you control which service consumers (AWS principals) can access your endpoint service. For example, you can grant permissions to specific principals that you trust and automatically accept all connection requests, or you can grant

permissions to a wider group of principals and manually accept specific connection requests that you trust.

You can configure your endpoint service to accept connection requests automatically. Otherwise, you must accept or reject them manually. If you do not accept a connection request, the service consumer can't access your endpoint service.

You can receive a notification when a connection request is accepted or rejected. For more information, see [the section called "Receive alerts for endpoint service events"](#).

Consideration

If you grant everyone permission to access the endpoint service and configure the endpoint service to accept all requests, your load balancer will be public even if it has no public IP address.

To modify the acceptance setting using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select the endpoint service.
4. Choose **Actions, Modify endpoint acceptance setting**.
5. Select or clear **Acceptance required**.
6. Choose **Save changes**

To modify the acceptance setting using the command line

- [modify-vpc-endpoint-service-configuration](#) (AWS CLI)
- [Edit-EC2VpcEndpointServiceConfiguration](#) (Tools for Windows PowerShell)

To accept or reject a connection request using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select the endpoint service.
4. From the **Endpoint connections** tab, select the endpoint connection.
5. To accept the connection request, choose **Actions, Accept endpoint connection request**. When prompted for confirmation, enter **accept** and then choose **Accept**.

6. To reject the connection request, choose **Actions, Reject endpoint connection request**. When prompted for confirmation, enter **reject** and then choose **Reject**.

To accept or reject a connection request using the command line

- [accept-vpc-endpoint-connections](#) or [reject-vpc-endpoint-connections](#) (AWS CLI)
- [Approve-EC2EndpointConnection](#) or [Deny-EC2EndpointConnection](#) (Tools for Windows PowerShell)

Change the load balancer association

You can change the load balancer that is associated with your endpoint service. You can't disassociate a load balancer if there are endpoints connected to your endpoint service.

To change the load balancers for your endpoint service using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select the endpoint service.
4. Choose **Actions, Associate or disassociate load balancers**.
5. Add or remove load balancers as needed.
6. Choose **Save changes**

To change the load balancers for your endpoint service using the command line

- [modify-vpc-endpoint-service-configuration](#) (AWS CLI)
- [Edit-EC2VpcEndpointServiceConfiguration](#) (Tools for Windows PowerShell)

Associate a private DNS name

You can associate a private DNS name with your endpoint service. After you associate a private DNS name, you must update the entry for the domain on your DNS server. Before service consumers can use the private DNS name, the service provider must verify that they own the domain. For more information, see [Manage DNS names](#).

To modify an endpoint service private DNS name using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select the endpoint service.
4. Choose **Actions, Modify private DNS name**.
5. Select **Associate a private DNS name with the service** and enter the private DNS name.
 - Domain names must use lowercase.
 - You can use wildcards in domain names (for example, ***.myexampleservice.com**).
6. Choose **Save changes**.
7. The private DNS name is ready for use by service consumers when the verification status is **verified**. If the verification status changes, new connection requests are denied but existing connections are not affected.

To modify an endpoint service private DNS name using the command line

- [modify-vpc-endpoint-service-configuration](#) (AWS CLI)
- [Edit-EC2VpcEndpointServiceConfiguration](#) (Tools for Windows PowerShell)

To initiate the domain verification process using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select the endpoint service.
4. Choose **Actions, Verify domain ownership for private DNS name**.
5. When prompted for confirmation, enter **verify** and then choose **Verify**.

To initiate the domain verification process using the command line

- [start-vpc-endpoint-service-private-dns-verification](#) (AWS CLI)
- [Start-EC2VpcEndpointServicePrivateDnsVerification](#) (Tools for Windows PowerShell)

Modify the supported IP address types

You can change the IP address types that are supported by your endpoint service.

Consideration

To enable your endpoint service to accept IPv6 requests, its Network Load Balancers must use the dualstack IP address type. The targets do not need to support IPv6 traffic. For more information, see [IP address type](#) in the *User Guide for Network Load Balancers*.

To modify the supported IP address types using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select the VPC endpoint service.
4. Choose **Actions, Modify supported IP address types**.
5. For **Supported IP address types**, do one of the following:
 - Select **IPv4** – Enable the endpoint service to accept IPv4 requests.
 - Select **IPv6** – Enable the endpoint service to accept IPv6 requests.
 - Select **IPv4 and IPv6** – Enable the endpoint service to accept both IPv4 and IPv6 requests.
6. Choose **Save changes**.

To modify the supported IP address types using the command line

- [modify-vpc-endpoint-service-configuration](#) (AWS CLI)
- [Edit-EC2VpcEndpointServiceConfiguration](#) (Tools for Windows PowerShell)

Manage tags

You can tag your resources to help you identify them or categorize them according to your organization's needs.

To manage tags for your endpoint service using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.

3. Select the VPC endpoint service.
4. Choose **Actions, Manage tags**.
5. For each tag to add, choose **Add new tag** and enter the tag key and tag value.
6. To remove a tag, choose **Remove** to the right of the tag key and value.
7. Choose **Save**.

To manage tags for your endpoint connections using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select the VPC endpoint service and then choose the **Endpoint connections** tab.
4. Select the endpoint connection and then choose **Actions, Manage tags**.
5. For each tag to add, choose **Add new tag** and enter the tag key and tag value.
6. To remove a tag, choose **Remove** to the right of the tag key and value.
7. Choose **Save**.

To manage tags for your endpoint service permissions using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select the VPC endpoint service and then choose the **Allow principals** tab.
4. Select the principal and then choose **Actions, Manage tags**.
5. For each tag to add, choose **Add new tag** and enter the tag key and tag value.
6. To remove a tag, choose **Remove** to the right of the tag key and value.
7. Choose **Save**.

To add and remove tags using the command line

- [create-tags](#) and [delete-tags](#) (AWS CLI)
- [New-EC2Tag](#) and [Remove-EC2Tag](#) (Tools for Windows PowerShell)

Manage DNS names for VPC endpoint services

Service providers can configure private DNS names for their endpoint services. When a service provider uses an existing public DNS name as the private DNS name for their endpoint service, then service consumers don't need to change any applications that use the existing public DNS name. Before you can configure a private DNS name for your endpoint service, you must prove that you own the domain by performing a domain ownership verification check.

Considerations

- An endpoint service can have only one private DNS name.
- You must not create an A record for the private DNS name, so that only servers in the service consumer VPC can resolve the private DNS name.
- Private DNS names are not supported for Gateway Load Balancer endpoints.
- To verify a domain, you must have a public hostname or a public DNS provider.
- You can verify the domain of a subdomain. For example, you can verify *example.com*, instead of *a.example.com*. As specified in [RFC 1034](#), each DNS label can have up to 63 characters and the whole domain name must not exceed a total length of 255 characters.

If you add an additional subdomain, you must verify the subdomain, or the domain. For example, let's say you had *a.example.com*, and verified *example.com*. You now add *b.example.com* as a private DNS name. You must verify *example.com* or *b.example.com* before service consumers can use the name.

Domain ownership verification

Your domain is associated with a set of domain name service (DNS) records that you manage through your DNS provider. A TXT record is a type of DNS record that provides additional information about your domain. It consists of a name and a value. As part of the verification process, you must add a TXT record to the DNS server for your public domain.

Domain ownership verification is complete when we detect the existence of the TXT record in your domain's DNS settings.

After you add a record, you can check the status of the domain verification process using the Amazon VPC console. In the navigation pane, choose **Endpoint services**. Select the endpoint service and check the value of **Domain verification status** in the **Details** tab. If domain verification

is pending, wait a few minutes and refresh the screen. If needed, you can initiate the verification process manually. Choose **Actions, Verify domain ownership for private DNS name**.

The private DNS name is ready for use by service consumers when the verification status is **verified**. If the verification status changes, new connection requests are denied but existing connections are not affected.

If the verification status is **failed**, see [the section called "Troubleshoot domain verification issues"](#).

Get the name and value

We provide you with the name and value that you use in the TXT record. For example, the information is available in the AWS Management Console. Select the endpoint service and see **Domain verification name** and **Domain verification value** on the **Details** tab for the endpoint service. You can also use the following [describe-vcpc-endpoint-service-configurations](#) AWS CLI command to retrieve information about the configuration of the private DNS name for the specified endpoint service.

```
aws ec2 describe-vcpc-endpoint-service-configurations \
  --service-ids vpce-svc-071afff70666e61e0 \
  --query ServiceConfigurations[*].PrivateDnsNameConfiguration
```

The following is example output. You'll use `Value` and `Name` when you create the TXT record.

```
[
  {
    "State": "pendingVerification",
    "Type": "TXT",
    "Value": "vpce:l6p0ERx1Tt45jevFw0Cp",
    "Name": "_6e86v84tqqqubxbwii1m"
  }
]
```

For example, suppose that your domain name is *example.com* and that `Value` and `Name` are as shown in the preceding example output. The following table is an example of the TXT record settings.

Name	Type	Value
_6e86v84tqqqubxbwi i1m.example.com	TXT	vpce:l6p0ERxlTt45jevFwOCp

We suggest that you use Name as the record subdomain because the base domain name might already be in use. However, if your DNS provider does not allow DNS record names to contain underscores, you can omit the "_6e86v84tqqqubxbwii1m" and simply use "example.com" in the TXT record.

After we verify "_6e86v84tqqqubxbwii1m.example.com", service consumers can use "example.com" or a subdomain (for example, "service.example.com" or "my.service.example.com").

Add a TXT record to your domain's DNS server

The procedure for adding TXT records to your domain's DNS server depends on who provides your DNS service. Your DNS provider might be Amazon Route 53 or another domain name registrar.

Amazon Route 53

Create a record for your public hosted zone. Use the following values:

- For **Record type**, choose **TXT**.
- For **TTL (seconds)**, enter **1800**.
- For **Routing policy**, choose **Simple routing**.
- For **Record name** enter the domain or subdomain.
- For **Value/Route traffic to**, enter the domain verification value.

For more information, see [Create records using the console](#) in the *Amazon Route 53 Developer Guide*.

General procedure

Go to the website for your DNS provider and sign in to your account. Find the page to update the DNS records for your domain. Add a TXT record with the name and value that we provided. It can take up to 48 hours for DNS record updates to take effect, but they often take effect much sooner.

For more specific directions, consult the documentation from your DNS provider. The following table provides links to the documentation for several common DNS providers. This list is not intended to be comprehensive, nor is it intended as a recommendation of the products or services provided by these companies.

DNS/Hosting provider	Documentation link
GoDaddy	Add a TXT record
Dreamhost	Adding custom DNS records
Cloudflare	Manage DNS records
HostGator	Manage DNS Records with HostGator/eNom
Namecheap	How do I add TXT/SPF/DKIM/DMARC records for my domain?
Names.co.uk	Changing your domain's DNS settings
Wix	Adding or Updating TXT Records in Your Wix Account

Check whether the TXT record is published

You can verify that your private DNS name domain ownership verification TXT record is published correctly to your DNS server using the following steps. You'll run the [nslookup](#) tool, which is available for Windows and Linux.

You'll query the DNS servers that serve your domain because those servers contain the most up-to-date information for your domain. Your domain information takes time to propagate to other DNS servers.

To verify that your TXT record is published to your DNS server

1. Find the name servers for your domain using the following command.

```
nslookup -type=NS example.com
```

The output lists the name servers that serve your domain. You'll query one of these servers in the next step.

2. Verify that the TXT record is correctly published using the following command, where *name_server* is one of the name servers that you found in the previous step.

```
nslookup -type=TXT _6e86v84tqqqubxbwii1m.example.com name_server
```

3. In the output of the previous step, verify that the string that follows `text =` matches the TXT value.

In our example, if the record is correctly published, the output includes the following.

```
_6e86v84tqqqubxbwii1m.example.com text = "vpce:l6p0ERx1Tt45jevFw0Cp"
```

Troubleshoot domain verification issues

If the domain verification process fails, the following information can help you troubleshoot issues.

- Check whether your DNS provider allows underscores in TXT record names. If your DNS provider does not allow underscores, you can omit the domain verification name (for example, "_6e86v84tqqqubxbwii1m") from the TXT record.
- Check whether your DNS provider appended the domain name to the end of the TXT record. Some DNS providers automatically append the name of your domain to the attribute name of the TXT record. To avoid this duplication of the domain name, add a period to the end of the domain name when you create the TXT record. This tells your DNS provider that it isn't necessary to append the domain name to the TXT record.
- Check whether your DNS provider modified the DNS record value to use only lowercase letters. We verify your domain only when there is a verification record with an attribute value that exactly matches the value that we provided. If the DNS provider changed your TXT record values to use only lowercase letters, contact them for assistance.
- You might need to verify your domain more than once because you're supporting multiple Regions or multiple AWS accounts. If your DNS provider doesn't allow you to have more than one TXT record with the same attribute name, check whether your DNS provider allows you to assign multiple attribute values to the same TXT record. For example, if your DNS is managed by Amazon Route 53, you can use the following procedure.
 1. In the Route 53 console, choose the TXT record that you created when you verified your domain in the first Region.
 2. For **Value**, go to the end of the existing attribute value, and then press Enter.

3. Add the attribute value for the additional Region, and then save the record set.

If your DNS provider doesn't allow you to assign multiple values to the same TXT record, you can verify the domain once with the value in the attribute name of the TXT record, and one other time with the value removed from the attribute name. However, you can only verify the same domain two times.

Receive alerts for endpoint service events

You can create a notification to receive alerts for specific events related to your endpoint service. For example, you can receive an email when a connection request is accepted or rejected.

Tasks

- [Create an SNS notification](#)
- [Add an access policy](#)
- [Add a key policy](#)

Create an SNS notification

Use the following procedure to create an Amazon SNS topic for the notifications and subscribe to the topic.

To create a notification for an endpoint service using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select the endpoint service.
4. From the **Notifications** tab, choose **Create notification**.
5. For **Notification ARN**, choose the ARN for the SNS topic that you created.
6. To subscribe to an event, select it from **Events**.
 - **Connect** – The service consumer created the interface endpoint. This sends a connection request to the service provider.
 - **Accept** – The service provider accepted the connection request.
 - **Reject** – The service provider rejected the connection request.

- **Delete** – The service consumer deleted the interface endpoint.

7. Choose **Create notification**.

To create a notification for an endpoint service using the command line

- [create-vpc-endpoint-connection-notification](#) (AWS CLI)
- [New-EC2VpcEndpointConnectionNotification](#) (Tools for Windows PowerShell)

Add an access policy

Add an access policy to the SNS topic that allows AWS PrivateLink to publish notifications on your behalf, such as the following. For more information, see [How do I edit my Amazon SNS topic's access policy?](#) Use the `aws:SourceArn` and `aws:SourceAccount` global condition keys to protect against the [confused deputy problem](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vpce.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:region:account-id:topic-name",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:ec2:region:account-id:vpc-endpoint-service/service-
id"
        },
        "StringEquals": {
          "aws:SourceAccount": "account-id"
        }
      }
    }
  ]
}
```

Add a key policy

If you're using encrypted SNS topics, the resource policy for the KMS key must trust AWS PrivateLink to call AWS KMS API operations. The following is an example key policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vpce.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:region:account-id:key/key-id",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:ec2:region:account-id:vpce-endpoint-service/service-id"
        },
        "StringEquals": {
          "aws:SourceAccount": "account-id"
        }
      }
    }
  ]
}
```

Delete an endpoint service

When you are finished with an endpoint service, you can delete it. You can't delete an endpoint service if there are any endpoints connected to the endpoint service that are in the available or pending-acceptance state.

Deleting an endpoint service does not delete the associated load balancer and does not affect the application servers registered with the load balancer target groups.

To delete an endpoint service using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select the endpoint service.
4. Choose **Actions, Delete endpoint services**.
5. When prompted for confirmation, enter **delete** and then choose **Delete**.

To delete an endpoint service using the command line

- [delete-vpc-endpoint-service-configurations](#) (AWS CLI)
- [Remove-EC2EndpointServiceConfiguration](#) (Tools for Windows PowerShell)

Identity and access management for AWS PrivateLink

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

Contents

- [Audience](#)
- [Authenticating with identities](#)
- [Managing access using policies](#)
- [How AWS PrivateLink works with IAM](#)
- [Identity-based policy examples for AWS PrivateLink](#)
- [Control access to VPC endpoints using endpoint policies](#)

Audience

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

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

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 (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 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. 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 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 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.
 - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see [Forward access sessions](#).
 - **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 AWS PrivateLink works with IAM

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

IAM features you can use with AWS PrivateLink

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

To get a high-level view of how AWS PrivateLink and other AWS services work with most IAM features, see [AWS services that work with IAM](#) in the *IAM User Guide*.

Identity-based policies for AWS PrivateLink

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*.

Identity-based policy examples for AWS PrivateLink

To view examples of AWS PrivateLink identity-based policies, see [Identity-based policy examples for AWS PrivateLink](#).

Resource-based policies within AWS PrivateLink

Supports resource-based policies	Yes
----------------------------------	-----

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*.

AWS PrivateLink service supports one type of resource-based policy, known as an *endpoint policy*. An endpoint policy controls which AWS principals can use the endpoint to access the endpoint service. For more information, see [the section called "Endpoint policies"](#).

Policy actions for AWS PrivateLink

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.

AWS PrivateLink shares its API namespace with Amazon EC2. Policy actions in AWS PrivateLink 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*"
```

To see a list of AWS PrivateLink actions, see [AWS PrivateLink actions](#) in the *Amazon EC2 API Reference*. For more information, see [Actions Defined by Amazon EC2](#) in the *Service Authorization Reference*.

Policy resources for AWS PrivateLink

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": "*" 
```

Policy condition keys for AWS PrivateLink

Supports service-specific policy condition keys Yes

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

The Condition element (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*.

The following condition keys are specific to AWS PrivateLink:

- `ec2:VpceServiceName`
- `ec2:VpceServiceOwner`
- `ec2:VpceServicePrivateDnsName`

To learn with which actions and resources you can use a condition key, see [Actions Defined by Amazon EC2](#).

ACLs in AWS PrivateLink

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 AWS PrivateLink

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 AWS PrivateLink

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 AWS PrivateLink

Supports forward access sessions (FAS)	Yes
--	-----

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to

complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see [Forward access sessions](#).

Service roles for AWS PrivateLink

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 AWS PrivateLink

Supports service-linked roles	No
-------------------------------	----

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

Identity-based policy examples for AWS PrivateLink

By default, users and roles don't have permission to create or modify AWS PrivateLink resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. 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.

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

For details about actions and resource types defined by AWS PrivateLink, including the format of the ARNs for each of the resource types, see [Actions, resources, and condition keys for Amazon EC2](#) in the *Service Authorization Reference*.

Examples

- [Control the use of VPC endpoints](#)
- [Control VPC endpoints creation based on the service owner](#)
- [Control the private DNS names that can be specified for VPC endpoint services](#)
- [Control the service names that can be specified for VPC endpoint services](#)

Control the use of VPC endpoints

By default, users do not have permission to work with endpoints. You can create an identity-based policy that grants users permission to create, modify, describe, and delete endpoints. The following is an example.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*VpcEndpoint*",
      "Resource": "*"
    }
  ]
}
```

For information about controlling access to services using VPC endpoints, see [the section called “Endpoint policies”](#).

Control VPC endpoints creation based on the service owner

You can use the `ec2:VpceServiceOwner` condition key to control what VPC endpoint can be created based on who owns the service (amazon, aws-marketplace, or the account ID). The following example grants permission to create VPC endpoints with the specified service owner. To use this example, substitute the Region, the account ID, and the service owner.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CreateVpcEndpoint",
      "Resource": [
```

```

        "arn:aws:ec2:region:account-id:vpc/*",
        "arn:aws:ec2:region:account-id:security-group/*",
        "arn:aws:ec2:region:account-id:subnet/*",
        "arn:aws:ec2:region:account-id:route-table/*"
    ]
},
{
    "Effect": "Allow",
    "Action": "ec2:CreateVpcEndpoint",
    "Resource": [
        "arn:aws:ec2:region:account-id:vpc-endpoint/*"
    ],
    "Condition": {
        "StringEquals": {
            "ec2:VpceServiceOwner": [
                "amazon"
            ]
        }
    }
}
]
}

```

Control the private DNS names that can be specified for VPC endpoint services

You can use the `ec2:VpceServicePrivateDnsName` condition key to control what VPC endpoint service can be modified or created based on the private DNS name associated with the VPC endpoint service. The following example grants permission to create a VPC endpoint service with the specified private DNS name. To use this example, substitute the Region, the account ID, and the private DNS name.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ModifyVpcEndpointServiceConfiguration",
                "ec2:CreateVpcEndpointServiceConfiguration"
            ],
            "Resource": [

```

```

        "arn:aws:ec2:region:account-id:vpc-endpoint-service/*"
    ],
    "Condition": {
        "StringEquals": {
            "ec2:VpceServicePrivateDnsName": [
                "example.com"
            ]
        }
    }
}
]
}

```

Control the service names that can be specified for VPC endpoint services

You can use the `ec2:VpceServiceName` condition key to control what VPC endpoint can be created based on the VPC endpoint service name. The following example grants permission to create a VPC endpoint with the specified service name. To use this example, substitute the Region, the account ID, and the service name.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CreateVpcEndpoint",
      "Resource": [
        "arn:aws:ec2:region:account-id:vpc/*",
        "arn:aws:ec2:region:account-id:security-group/*",
        "arn:aws:ec2:region:account-id:subnet/*",
        "arn:aws:ec2:region:account-id:route-table/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateVpcEndpoint",
      "Resource": [
        "arn:aws:ec2:region:account-id:vpc-endpoint/*"
      ],
      "Condition": {
        "StringEquals": {

```

```
        "ec2:VpceServiceName": [
            "com.amazonaws.region.s3"
        ]
    }
}
]
```

Control access to VPC endpoints using endpoint policies

An endpoint policy is a resource-based policy that you attach to a VPC endpoint to control which AWS principals can use the endpoint to access an AWS service.

An endpoint policy does not override or replace identity-based policies or resource-based policies. For example, if you're using an interface endpoint to connect to Amazon S3, you can also use Amazon S3 bucket policies to control access to buckets from specific endpoints or specific VPCs.

Contents

- [Considerations](#)
- [Default endpoint policy](#)
- [Policies for interface endpoints](#)
- [Principals for gateway endpoints](#)
- [Update a VPC endpoint policy](#)

Considerations

- An endpoint policy is a JSON policy document that uses the IAM policy language. It must contain a [Principal](#) element. The size of an endpoint policy cannot exceed 20,480 characters, including white space.
- When you create an interface or gateway endpoint for an AWS service, you can attach a single endpoint policy to the endpoint. You can [update the endpoint policy](#) at any time. If you don't attach an endpoint policy, we attach the [default endpoint policy](#).
- Not all AWS services support endpoint policies. If an AWS service doesn't support endpoint policies, we allow full access to any endpoint for the service. For more information, see [the section called "View endpoint policy support"](#).

- When you create a VPC endpoint for an endpoint service other than an AWS service, we allow full access to the endpoint.

Default endpoint policy

The default endpoint policy grants full access to the endpoint.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "*",
      "Resource": "*"
    }
  ]
}
```

Policies for interface endpoints

For example endpoint policies for AWS services, see [the section called "Services that integrate"](#).

The first column in the table contains links to AWS PrivateLink documentation for each AWS service. If an AWS service supports endpoint policies, its documentation includes example endpoint policies.

Principals for gateway endpoints

With gateway endpoints, the `Principal` element must be set to `*`. To specify a principal, use the `aws:PrincipalArn` condition key.

```
"Condition": {
  "StringEquals": {
    "aws:PrincipalArn": "arn:aws:iam::123456789012:user/endpointuser"
  }
}
```

If you specify the principal in the following format, access is granted to the AWS account root user only, not all users and roles for the account.

```
"AWS": "account_id"
```

For example endpoint policies for gateway endpoints, see the following:

- [Endpoints for Amazon S3](#)
- [Endpoints for DynamoDB](#)

Update a VPC endpoint policy

Use the following procedure to update an endpoint policy for an AWS service. After you update an endpoint policy, it can take a few minutes for the changes to take effect.

To update an endpoint policy using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Select the VPC endpoint.
4. Choose **Actions, Manage policy**.
5. Choose **Full Access** to allow full access to the service, or choose **Custom** and attach a custom policy.
6. Choose **Save**.

To update an endpoint policy using the command line

- [modify-vpc-endpoint](#) (AWS CLI)
- [Edit-EC2VpcEndpoint](#) (Tools for Windows PowerShell)

CloudWatch metrics for AWS PrivateLink

AWS PrivateLink publishes data points to Amazon CloudWatch for your interface endpoints, Gateway Load Balancer endpoints, and endpoint services. CloudWatch enables you to retrieve statistics about those data points as an ordered set of time series data, known as *metrics*. Think of a metric as a variable to monitor, and the data points as the values of that variable over time. Each data point has an associated timestamp and an optional unit of measurement.

You can use metrics to verify that your system is performing as expected. For example, you can create a CloudWatch alarm to monitor a specified metric and initiate an action (such as sending a notification to an email address) if the metric goes outside what you consider an acceptable range.

Metrics are published for all interface endpoints, Gateway Load Balancer endpoints, and endpoint services. They are not published for gateway endpoints. By default, AWS PrivateLink sends metrics to CloudWatch in one-minute intervals, at no additional cost.

For more information, see the [Amazon CloudWatch User Guide](#).

Contents

- [Endpoint metrics and dimensions](#)
- [Endpoint service metrics and dimensions](#)
- [View the CloudWatch metrics](#)
- [Use built-in Contributor Insights rules](#)

Endpoint metrics and dimensions

The `AWS/PrivateLinkEndpoints` namespace includes the following metrics for interface endpoints and Gateway Load Balancer endpoints.

Metric	Description
<code>ActiveConnections</code>	<p>The number of concurrent active connections. This includes connections in the <code>SYN_SENT</code> and <code>ESTABLISHED</code> states.</p> <p>Reporting criteria: The endpoint received traffic during the one-minute period.</p>

Metric	Description
	<p>Statistics: The most useful statistics are Average, Maximum, and Minimum.</p> <p>Dimensions</p> <ul style="list-style-type: none"> • Endpoint Type, Service Name, VPC Endpoint Id, VPC Id • Endpoint Type, Service Name, Subnet Id, VPC Endpoint Id, VPC Id
BytesProcessed	<p>The number of bytes exchanged between endpoints and endpoint services, aggregated in both directions. This is the number of bytes billed to the owner of the endpoint. The bill displays this value in GB.</p> <p>Reporting criteria: The endpoint received traffic during the one-minute period.</p> <p>Statistics: The most useful statistics are Average, Sum, Maximum, and Minimum.</p> <p>Dimensions</p> <ul style="list-style-type: none"> • Endpoint Type, Service Name, VPC Endpoint Id, VPC Id • Endpoint Type, Service Name, Subnet Id, VPC Endpoint Id, VPC Id

Metric	Description
NewConnections	<p>The number of new connections established through the endpoint.</p> <p>Reporting criteria: The endpoint received traffic during the one-minute period.</p> <p>Statistics: The most useful statistics are Average, Sum, Maximum, and Minimum.</p> <p>Dimensions</p> <ul style="list-style-type: none">• Endpoint Type, Service Name, VPC Endpoint Id, VPC Id• Endpoint Type, Service Name, Subnet Id, VPC Endpoint Id, VPC Id
PacketsDropped	<p>The number of packets dropped by the endpoint. This metric might not capture all packet drops. Increasing values could indicate that the endpoint or endpoint service is unhealthy.</p> <p>Reporting criteria: The endpoint received traffic during the one-minute period.</p> <p>Statistics: The most useful statistics are Average, Sum, and Maximum.</p> <p>Dimensions</p> <ul style="list-style-type: none">• Endpoint Type, Service Name, VPC Endpoint Id, VPC Id• Endpoint Type, Service Name, Subnet Id, VPC Endpoint Id, VPC Id

Metric	Description
RstPacketsReceived	<p>The number of RST packets received by the endpoint. Increasing values could indicate that the endpoint service is unhealthy.</p> <p>Reporting criteria: The endpoint received traffic during the one-minute period.</p> <p>Statistics: The most useful statistics are Average, Sum, and Maximum.</p> <p>Dimensions</p> <ul style="list-style-type: none"> Endpoint Type, Service Name, VPC Endpoint Id, VPC Id Endpoint Type, Service Name, Subnet Id, VPC Endpoint Id, VPC Id

To filter these metrics, use the following dimensions.

Dimension	Description
Endpoint Type	Filters the metric data by endpoint type (Interface GatewayLoadBalancer).
Service Name	Filters the metric data by service name.
Subnet Id	Filters the metric data by subnet.
VPC Endpoint Id	Filters the metric data by VPC endpoint.
VPC Id	Filters the metric data by VPC.

Endpoint service metrics and dimensions

The AWS/PrivateLinkServices namespace includes the following metrics for endpoint services.

Metric	Description
ActiveConnections	<p>The maximum number of active connections from clients to targets through the endpoints. Increasing values could indicate the need to add targets to the load balancer.</p> <p>Reporting criteria: An endpoint connected to the endpoint service sent traffic during the one-minute period.</p> <p>Statistics: The most useful statistics are Average and Maximum.</p> <p>Dimensions</p> <ul style="list-style-type: none"> • Service Id • Az, Service Id • Load Balancer Arn, Service Id • Az, Load Balancer Arn, Service Id • Service Id, VPC Endpoint Id
BytesProcessed	<p>The number of bytes exchanged between endpoint services and endpoints, in both directions.</p> <p>Reporting criteria: An endpoint connected to the endpoint service sent traffic during the one-minute period.</p> <p>Statistics: The most useful statistics are Average, Sum, and Maximum.</p> <p>Dimensions</p> <ul style="list-style-type: none"> • Service Id • Az, Service Id • Load Balancer Arn, Service Id • Az, Load Balancer Arn, Service Id • Service Id, VPC Endpoint Id
EndpointsCount	<p>The number of endpoints connected to the endpoint service.</p>

Metric	Description
	<p>Reporting criteria: There is a nonzero value during the five-minute period.</p> <p>Statistics: The most useful statistics are Average and Maximum.</p> <p>Dimensions</p> <ul style="list-style-type: none"> • Service Id
NewConnections	<p>The number of new connections established from clients to targets through the endpoints. Increasing values could indicate the need to add targets to the load balancer.</p> <p>Reporting criteria: An endpoint connected to the endpoint service sent traffic during the one-minute period.</p> <p>Statistics: The most useful statistics are Average, Sum, and Maximum.</p> <p>Dimensions</p> <ul style="list-style-type: none"> • Service Id • Az, Service Id • Load Balancer Arn, Service Id • Az, Load Balancer Arn, Service Id • Service Id, VPC Endpoint Id

Metric	Description
RstPacketsSent	<p>The number of RST packets sent to endpoints by the endpoint service. Increasing values could indicate that there are unhealthy targets.</p> <p>Reporting criteria: An endpoint connected to the endpoint service sent traffic during the one-minute period.</p> <p>Statistics: The most useful statistics are Average, Sum, and Maximum.</p> <p>Dimensions</p> <ul style="list-style-type: none"> • Service Id • Az, Service Id • Load Balancer Arn, Service Id • Az, Load Balancer Arn, Service Id • Service Id, VPC Endpoint Id

To filter these metrics, use the following dimensions.

Dimension	Description
Az	Filters the metric data by Availability Zone.
Load Balancer Arn	Filters the metric data by load balancer.
Service Id	Filters the metric data by endpoint service.
VPC Endpoint Id	Filters the metric data by VPC endpoint.

View the CloudWatch metrics

You can view these CloudWatch metrics using the Amazon VPC console, the CloudWatch console, or the AWS CLI as follows.

To view metrics using the Amazon VPC console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**. Select your endpoint and then choose the **Monitoring** tab.
3. In the navigation pane, choose **Endpoint services**. Select your endpoint service and then choose the **Monitoring** tab.

To view metrics using the CloudWatch console

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. Select the **AWS/PrivateLinkEndpoints** namespace.
4. Select the **AWS/PrivateLinkServices** namespace.

To view metrics using the AWS CLI

Use the following [list-metrics](#) command to list the available metrics for interface endpoints and Gateway Load Balancer endpoints:

```
aws cloudwatch list-metrics --namespace AWS/PrivateLinkEndpoints
```

Use the following [list-metrics](#) command to list the available metrics for endpoint services:

```
aws cloudwatch list-metrics --namespace AWS/PrivateLinkServices
```

Use built-in Contributor Insights rules

AWS PrivateLink provides built-in Contributor Insights rules for your endpoint services to help you find which endpoints are the largest contributors to each supported metric. For more information, see [Contributor Insights](#) in the *Amazon CloudWatch User Guide*.

AWS PrivateLink provides the following rules:

- `VpcEndpointService-ActiveConnectionsByEndpointId-v1` – Ranks endpoints by the number of active connections.

- `VpcEndpointService-BytesByEndpointId-v1` – Ranks endpoints by the number of bytes processed.
- `VpcEndpointService-NewConnectionsByEndpointId-v1` – Ranks endpoints by the number of new connections.
- `VpcEndpointService-RstPacketsByEndpointId-v1` – Ranks endpoints by the number of RST packets sent to endpoints.

Before you can use a built-in rule, you must enable it. After you enable a rule, it starts collecting contributor data. For information about the charges for Contributor Insights, see [Amazon CloudWatch Pricing](#).

You must have the following permissions to use Contributor Insights:

- `cloudwatch:DeleteInsightRules` – To delete Contributor Insights rules.
- `cloudwatch:DisableInsightRules` – To disable Contributor Insights rules.
- `cloudwatch:GetInsightRuleReport` – To get the data.
- `cloudwatch:ListManagedInsightRules` – To list the available Contributor Insights rules.
- `cloudwatch:PutManagedInsightRules` – To enable Contributor Insights rules.

Tasks

- [Enable Contributor Insights rules](#)
- [Disable Contributor Insights rules](#)
- [Delete Contributor Insights rules](#)

Enable Contributor Insights rules

Use the following procedures to enable the built-in rules for AWS PrivateLink using either the AWS Management Console or the AWS CLI.

To enable the Contributor Insights rules for AWS PrivateLink using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select your endpoint service.
4. On the **Contributor Insights** tab, choose **Enable**.

5. (Optional) By default, all rules are enabled. To enable only specific rules, select the rules that should not be enabled and then choose **Actions, Disable rule**. When prompted for confirmation, choose **Disable**.

To enable the Contributor Insights rules for AWS PrivateLink using the AWS CLI

1. Use the [list-managed-insight-rules](#) command as follows to enumerate the available rules. For the `--resource-arn` option, specify the ARN of your endpoint service.

```
aws cloudwatch list-managed-insight-rules --resource-arn
arn:aws:ec2:region:account-id:vpc-endpoint-service/vpc-svc-0123456789EXAMPLE
```

2. In the output of the `list-managed-insight-rules` command, copy the name of the template from the `TemplateName` field. The following is an example of this field.

```
"TemplateName": "VpcEndpointService-NewConnectionsByEndpointId-v1"
```

3. Use the [put-managed-insight-rules](#) command as follows to enable the rule. You must specify the template name and the ARN of your endpoint service.

```
aws cloudwatch put-managed-insight-rules --managed-rules
TemplateName=VpcEndpointService-NewConnectionsByEndpointId-v1,
ResourceARN=arn:aws:ec2:region:account-id:vpc-endpoint-service/vpc-
svc-0123456789EXAMPLE
```

Disable Contributor Insights rules

You can disable the built-in rules for AWS PrivateLink at any time. After you disable a rule, it stops collecting contributor data, but existing contributor data is kept until it is 15 days old. After you disable a rule, you can enable it again to resume collecting contributor data.

To disable the Contributor Insights rules for AWS PrivateLink using the console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoint services**.
3. Select your endpoint service.
4. On the **Contributor Insights** tab, choose **Disable all** to disable all rules. Alternatively, expand the **Rules** panel, select the rules to disable, and then choose **Actions, Disable rule**

5. When prompted for confirmation, choose **Disable**.

To disable the Contributor Insights rules for AWS PrivateLink using the AWS CLI

Use the [disable-insight-rules](#) command to disable a rule.

Delete Contributor Insights rules

Use the following procedures to delete the built-in rules for AWS PrivateLink using either the AWS Management Console or the AWS CLI. After you delete a rule, it stops collecting contributor data and we delete the existing contributor data.

To delete Contributor Insights rules for AWS PrivateLink using the console

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Insights, Contributor Insights**.
3. Expand the **Rules** panel and select the rules.
4. Choose **Actions, Delete rule**.
5. When prompted for confirmation, choose **Delete**.

To delete Contributor Insights rules for AWS PrivateLink using the AWS CLI

Use the [delete-insight-rules](#) command to delete a rule.

AWS PrivateLink quotas

The following tables list the quotas, formerly referred to as limits, for AWS PrivateLink resources per Region for your account. Unless indicated otherwise, you can request an increase for these quotas. For more information, see [Requesting a quota increase](#) in the *Service Quotas User Guide*.

If you request a quota increase that applies per resource, we increase the quota for all resources in the Region.

Name	Default	Adjustable	Comments
Interface and Gateway Load Balancer endpoints per VPC	50	Yes	This is a combined quota for interface endpoints and Gateway Load Balancer endpoints
Gateway VPC endpoints per Region	20	Yes	You can create up to 255 gateway endpoints per VPC
Characters per VPC endpoint policy	20,480	No	The maximum size of a VPC endpoint policy, including white space

The following considerations apply to traffic that passes through a VPC endpoint:

- By default, each VPC endpoint can support a bandwidth of up to 10 Gbps per Availability Zone, and automatically scales up to 100 Gbps. The maximum bandwidth for a VPC endpoint, when distributing the load across all Availability Zones, is the number of Availability Zones multiplied by 100 Gbps. If your application needs higher throughput, contact AWS support.
- The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed through a VPC endpoint. The larger the MTU, the more data that can be passed in a single packet. A VPC endpoint supports an MTU of 8500 bytes. Packets with a size larger than 8500 bytes that arrive at the VPC endpoint are dropped.
- Path MTU Discovery (PMTUD) is not supported. VPC endpoints do not generate the following ICMP message: Destination Unreachable: Fragmentation needed and Don't Fragment was Set (Type 3, Code 4).

- VPC endpoints enforce Maximum Segment Size (MSS) clamping for all packets. For more information, see [RFC879](#).

Document history for AWS PrivateLink

The following table describes the releases for AWS PrivateLink.

Change	Description	Date
Designated IP addresses	You can specify the IP addresses for your endpoint network interfaces when you create or modify your VPC endpoint.	August 17, 2023
IPv6 support	You can configure your Gateway Load Balancer endpoint services and Gateway Load Balancer endpoints to support both IPv4 and IPv6 addresses or only IPv6 addresses.	December 12, 2022
Contributor Insights	You can use built-in Contributor or Insights rules to identify specific endpoints that are the top contributors to the CloudWatch metrics for AWS PrivateLink.	August 18, 2022
IPv6 support	Service providers can enable their endpoint service to accept IPv6 requests, even if their backend services support only IPv4. If an endpoint service accepts IPv6 requests, service consumers can enable IPv6 support for their interface endpoints	May 11, 2022

so that they can access the endpoint service over IPv6.

[CloudWatch metrics](#)

AWS PrivateLink publishes CloudWatch metrics for your interface endpoints, Gateway Load Balancer endpoints, and endpoint services.

January 27, 2022

[Gateway Load Balancer endpoints](#)

You can create a Gateway Load Balancer endpoint in your VPC to route traffic to a VPC endpoint service that you've configured using a Gateway Load Balancer.

November 10, 2020

[VPC endpoint policies](#)

You can attach an IAM policy to an interface VPC endpoint for an AWS service to control access to the service.

March 23, 2020

[Condition keys for VPC endpoints and endpoint services](#)

You can use EC2 condition keys to control access to VPC endpoints and endpoint services.

March 6, 2020

[Tag VPC endpoints and endpoint services on creation](#)

You can add tags when you create VPC endpoints and endpoint services.

February 5, 2020

[Private DNS names](#)

You can access AWS PrivateLink based services from within your VPC using private DNS names.

January 6, 2020

<u>VPC endpoint services</u>	You can create your own endpoints services and enable other AWS accounts and users to connect to your service through an interface VPC endpoint. You can offer your endpoint services for subscription in the AWS Marketplace.	November 28, 2017
<u>Interface VPC endpoints for AWS services</u>	You can create an interface endpoint to connect to AWS services that integrate with AWS PrivateLink without using an internet gateway or NAT device.	November 8, 2017
<u>VPC endpoints for DynamoDB</u>	You can create a gateway VPC endpoint to access Amazon DynamoDB from your VPC without using an internet gateway or NAT device.	August 16, 2017
<u>VPC endpoints for Amazon S3</u>	You can create a gateway VPC endpoint to access Amazon S3 from your VPC without using an internet gateway or NAT device.	May 11, 2015