AWS PrivateLink

Amazon Virtual Private Cloud



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

Amazon Virtual Private Cloud: AWS PrivateLink

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

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

Table of Contents

What is AWS PrivateLink? ... 1
Use cases ... 1
Work with VPC endpoints ... 3
Pricing ... 3
Concepts ... 3
Architecture diagram ... 4
Providers ... 4
Service or resource consumers ... 6
AWS PrivateLink connections ... 9
Private hosted zones ... 9
Get started ... 10
Step 1: Create a VPC with subnets ... 11
Step 2: Launch the instances ... 11
Step 3: Test CloudWatch access ... 13
Step 4: Create a VPC endpoint to access CloudWatch ... 14
Step 5: Test the VPC endpoint ... 14
Step 6: Clean up ... 15
Access AWS services ... 16
Overview ... 17
DNS hostnames ... 18
DNS resolution ... 20
Private DNS ... 20
Subnets and Availability Zones ... 21
IP address types ... 24
DNS record IP type ... 25
Services that integrate ... 26
View available AWS service names ... 51
View information about a service ... 51
View endpoint policy support ... 53
View IPv6 support ... 55
Cross-region enabled AWS services ... 55
View available AWS service names ... 51
Permissions and Considerations ... 57
Create an interface endpoint to an AWS service in another Region ... 58

Create an interface endpoint ... 58
Prerequisites ... 59
Create a VPC endpoint ... 59
Shared subnets ... 61
ICMP ... 61
Configure an interface endpoint ... 61
Add or remove subnets ... 62
Associate security groups ... 62
Edit the VPC endpoint policy ... 63
Enable private DNS names ... 64
Manage tags ... 64
Receive alerts for interface endpoint events ... 65
Create an SNS notification ... 65
Add an access policy ... 66
Add a key policy ... 67
Delete an interface endpoint ... 68
Gateway endpoints ... 68
Overview ... 69
Routing ... 71
Security ... 72
IP address type ... 72
DNS record IP type ... 73
Endpoints for Amazon S3 ... 74
Endpoints for DynamoDB ... 86
Access SaaS products ... 94
Overview ... 94
Create an interface endpoint ... 95
Share your services ... 97
Overview ... 97
DNS hostnames ... 98
Private DNS ... 99
Subnets and Availability Zones ... 99
Cross-Region access ... 99
IP address types ... 101
Create an endpoint service ... 102
Considerations ... 102

Prerequisites ... 103
Create an endpoint service ... 104
Make your endpoint service available to service consumers ... 105
Connect to an endpoint service as the service consumer ... 106
Configure an endpoint service ... 107
Manage permissions ... 107
Accept or reject connection requests ... 109
Manage load balancers ... 110
Associate a private DNS name ... 111
Modify the supported Regions ... 112
Modify the supported IP address types ... 113
Manage tags ... 114
Manage DNS names ... 115
Domain ownership verification ... 116
Get the name and value ... 116
Add a TXT record to your domain's DNS server ... 117
Check whether the TXT record is published ... 118
Troubleshoot domain verification issues ... 119
Receive alerts for endpoint service events ... 120
Create an SNS notification ... 120
Add an access policy ... 121
Add a key policy ... 122
Delete an endpoint service ... 123
Access virtual appliances ... 124
Overview ... 124
IP address types ... 126
Routing ... 127
Create a Gateway Load Balancer endpoint service ... 128
Considerations ... 128
Prerequisites ... 129
Create the endpoint service ... 129
Make your endpoint service available ... 130
Create a Gateway Load Balancer endpoint ... 130
Considerations ... 131
Prerequisites ... 132
Create the endpoint ... 132

Configure routing ... 133
Manage tags ... 134
Delete the endpoint ... 135
Access VPC resources ... 136
Overview ... 136
Considerations ... 137
DNS hostnames ... 137
DNS resolution ... 139
Private DNS ... 139
Subnets and Availability Zones ... 139
IP address types ... 139
Create a resource endpoint ... 140
Prerequisites ... 140
Create a VPC resource endpoint ... 141
Manage resource endpoints ... 142
Delete an endpoint ... 142
Update an endpoint ... 142
Access network segments ... 144
Overview ... 144
Considerations ... 145
GENEVE encapsulation ... 145
DNS hostnames ... 145
DNS resolution ... 145
Private DNS ... 146
Subnets and Availability Zones ... 146
IP address types ... 146
Create a tunnel endpoint ... 147
Prerequisites ... 147
Create a tunnel VPC endpoint ... 147
Access service networks ... 149
Overview ... 150
DNS hostnames ... 150
DNS resolution ... 151
Private DNS ... 151
Subnets and Availability Zones ... 152
IP address types ... 152

Create a service-network endpoint ... 153
Prerequisites ... 153
Create a service network endpoint ... 154
Manage service-network endpoints ... 155
Delete an endpoint ... 155
Update a service-network endpoint ... 155
Resource configuration ... 157
Types of resource configurations ... 157
Resource gateway ... 158
Custom domain names for resource providers ... 158
Custom domain names for resource consumers ... 159
Custom domain names for service network owners ... 161
Resource definition ... 161
Protocol ... 162
Port ranges ... 162
Accessing resources ... 162
Association with service network type ... 163
Types of service networks ... 163
Sharing resource configurations through AWS RAM ... 164
Monitoring ... 164
Create a resource configuration ... 164
Manage associations ... 166
Manage service network resource associations ... 167
Manage resource VPC endpoint associations ... 168
Resource gateway ... 158
Considerations ... 170
Security groups ... 171
IP address types ... 171
IPv4 addresses per ENI ... 172
Resource Config DNS Resolution ... 172
Create a resource gateway ... 173
Delete a resource gateway ... 174
Identity and access management ... 175
Audience ... 175
Authenticating with identities ... 176
AWS account root user ... 176

Federated identity ... 176
IAM users and groups ... 176
IAM roles ... 177
Managing access using policies ... 177
Identity-based policies ... 177
Resource-based policies ... 178
Other policy types ... 178
Multiple policy types ... 178
How AWS PrivateLink works with IAM ... 178
Identity-based policies ... 179
Resource-based policies ... 180
Policy actions ... 180
Policy resources ... 181
Policy condition keys ... 181
ACLs ... 182
ABAC ... 182
Temporary credentials ... 182
Principal permissions ... 182
Service roles ... 183
Service-linked roles ... 183
Identity-based policy examples ... 183
Control the use of VPC endpoints ... 184
Control VPC endpoints creation based on the service owner ... 184
Control the private DNS names that can be specified for VPC endpoint services ... 185
Control the service names that can be specified for VPC endpoint services ... 186
Endpoint policies ... 187
Considerations ... 187
Default endpoint policy ... 188
Policies for interface endpoints ... 188
Principals for gateway endpoints ... 189
Update a VPC endpoint policy ... 189
AWS managed policies ... 190
Policy updates ... 190
CloudWatch metrics ... 191
Endpoint metrics and dimensions ... 191
Endpoint service metrics and dimensions ... 194

View the CloudWatch metrics 197
Use built-in Contributor Insights rules 198
Enable Contributor Insights rules 199
Disable Contributor Insights rules 200
Delete Contributor Insights rules 201
Quotas 202
Document history 204

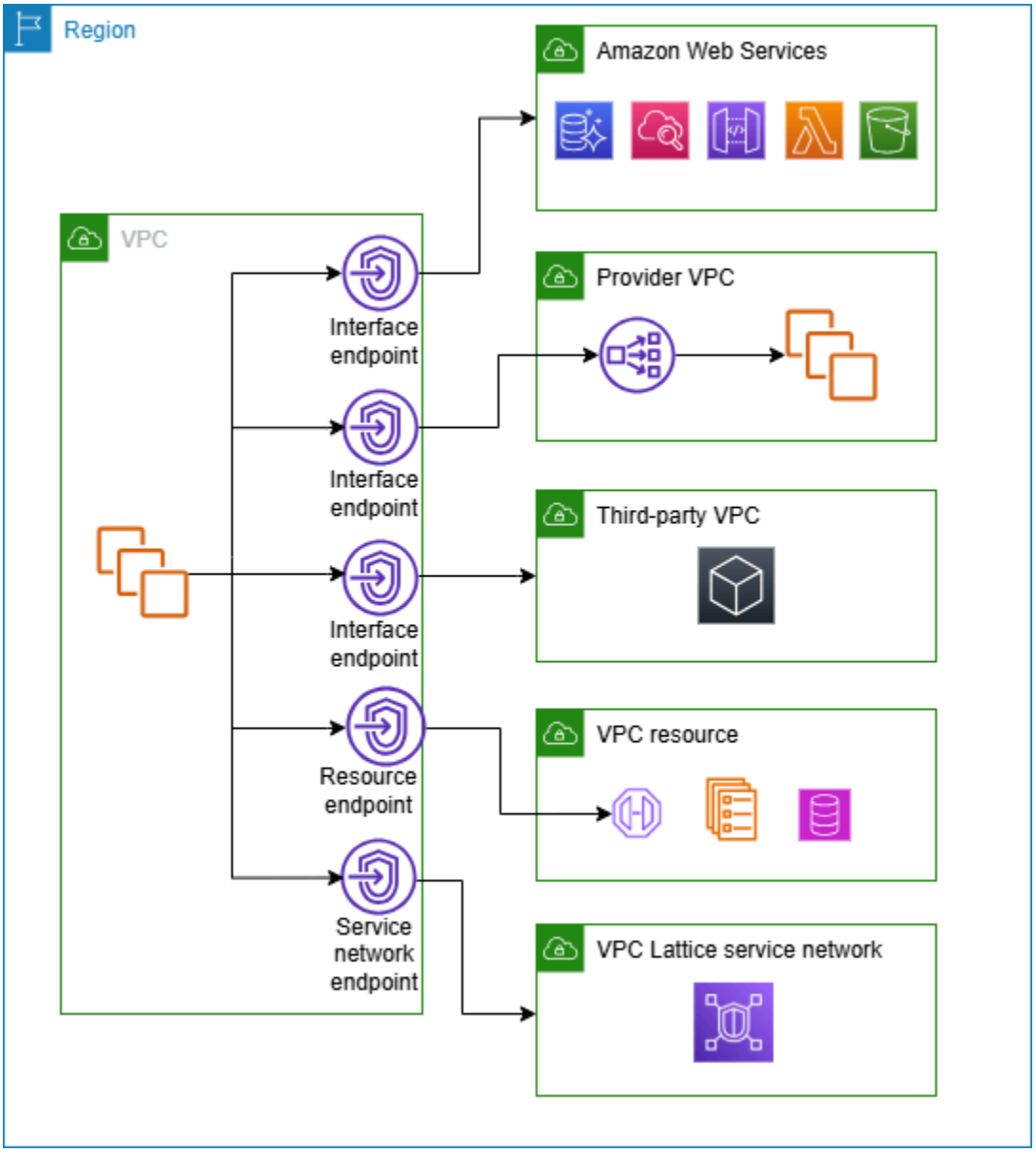
What is AWS PrivateLink?

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

Use cases

You can create AWS PrivateLink-based VPC endpoints to access services, resources, and network segments across VPC and account boundaries. 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 Amazon EC2 instances in a private subnet and five VPC endpoints - three interface VPC endpoints, a resource VPC endpoint and a service-network VPC endpoint. The first interface VPC endpoint connects to an AWS service. The second interface VPC endpoint connects to a service hosted by another AWS account (a VPC endpoint service). The third interface VPC endpoint connects to an AWS Marketplace partner service. The resource VPC endpoint connects to a database. The service network VPC endpoint connects to a service network.



Learn more

- 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. Open the Amazon VPC console and choose **Endpoints** or **Endpoint services**.
- **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*.
- **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 Tools to Build on AWS.
- **Query API** — Provides low-level API actions that you call using HTTPS requests. Using the Query API is the most direct way to access 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 allow the clients in your VPC to connect to destinations 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 clients in your VPC to connect to services and resources in other VPCs using private IP addresses, as if those services and resources were hosted directly in your VPC.

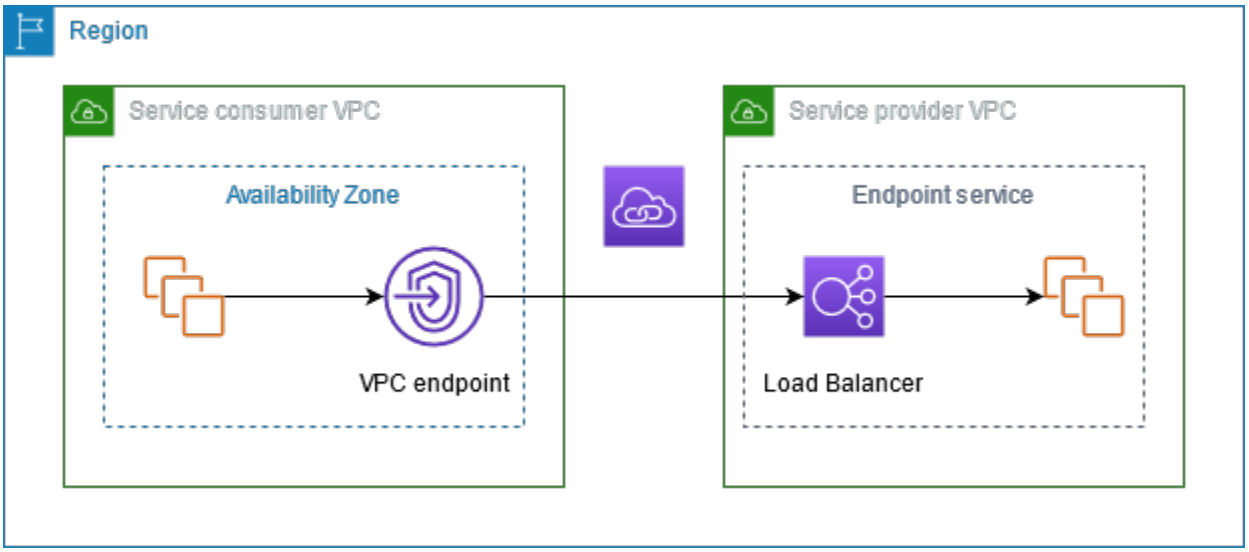
The following are important concepts to understand as you get started using AWS PrivateLink.

Contents

- Architecture diagram
- Providers
- Service or resource consumers
- AWS PrivateLink connections
- Private hosted zones

Architecture diagram

The following diagram provides a high-level overview of how AWS PrivateLink works. Consumers create VPC endpoints to connect to endpoint services and resources that are hosted by providers.



Providers

Understand the concepts related to a provider.

Service provider

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.

Resource provider

The owner of a resource, for example a database or an Amazon EC2 instance, is the resource provider. Resource providers include AWS services, AWS Partners, and other AWS accounts. Resource providers can host their resources in VPCs or on-premises.

Concepts

- Endpoint services
- Service names
- Service states
- Resource configuration
- Resource gateway

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.

Resource configuration

The resource provider creates a *resource configuration* to share a resource. A resource configuration is a logical object that represents a single resource such as a database, a group of resources, or a segment of a network (a set of CIDR ranges). A resource can be an IP address, a domain-name target, or an Amazon Relational Database Service (Amazon RDS) database.

When sharing with other accounts, the resource provider must share the resource through a AWS Resource Access Manager (AWS RAM) resource share to allow specific AWS principals in the other account to connect to the resource through a resource VPC endpoint.

Resource configurations can be associated with a service network which principals connect to through a service-network VPC endpoint.

Resource gateway

A resource gateway is a point of ingress into a VPC from where a resource is being shared. The provider creates a resource gateway to share resources from the VPC.

Service or resource consumers

The user of a service or resource is a *consumer*. Consumers can access endpoint services and resources from their VPCs or from on-premises.

Concepts

- VPC endpoints
- Endpoint network interfaces
- Endpoint policies
- Endpoint states

VPC endpoints

A consumer creates a *VPC endpoint* to connect their VPC to an endpoint service or resource. A consumer must specify the endpoint service, resource, or service network when creating a VPC endpoint. There are multiple types of VPC endpoints. You must create the type of VPC endpoint that you require.

- `Interface` - Create an *interface endpoint* to send TCP or UDP 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.
- `Resource` - Create a *resource endpoint* to access a resource that was shared with you and resides in another VPC. A resource endpoint lets you privately and securely access resources such as a database, an Amazon EC2 instance, an application endpoint, a domain-name target, or an IP address that may be in a private subnet in another VPC or in an on premise environment. Resource endpoints don't require a load balancer, and lets you access the resource directly.
- `Tunnel` - Create a *tunnel endpoint* to access a network segment that was shared with you from another VPC or account. A tunnel endpoint lets you privately and securely tunnel into a network segment (a list of CIDR ranges) and access any resource located in the segment. Tunnel endpoints require you to encapsulate application traffic in GENEVE.
- `Service network` - Create a *service-network endpoint* to access a service network that you created or was shared with you. You can use a single service-network endpoint to privately and securely access multiple resources and services that are associated to a service network.

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, resource, or service network. 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.

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 an interface 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.

The AWS PrivateLink API returns the possible states using camel case.

AWS PrivateLink connections

Traffic from your VPC is sent to an endpoint service or resource using a connection between the VPC endpoint and the endpoint service or resource. Traffic between a VPC endpoint and an endpoint service or resource 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. A resource owner or service network owner shares a resource configuration or service network with consumers through AWS Resource Access Manager so that consumers can access the resource or service network.

With interface VPC endpoints, consumers can use endpoint policies to control which IAM principals can use a VPC endpoint to access an endpoint service or resource.

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.

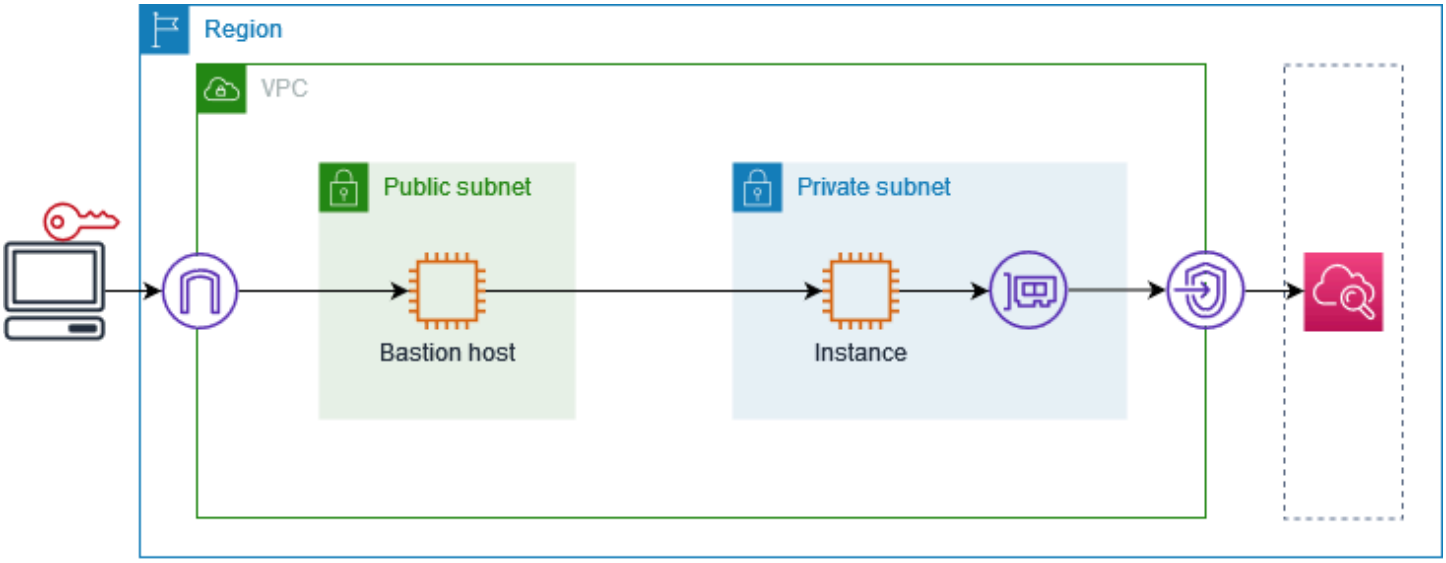
You can configure Amazon Route 53 to route domain traffic to a VPC endpoint. For more information, see Routing traffic to a VPC endpoint using your domain name.

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

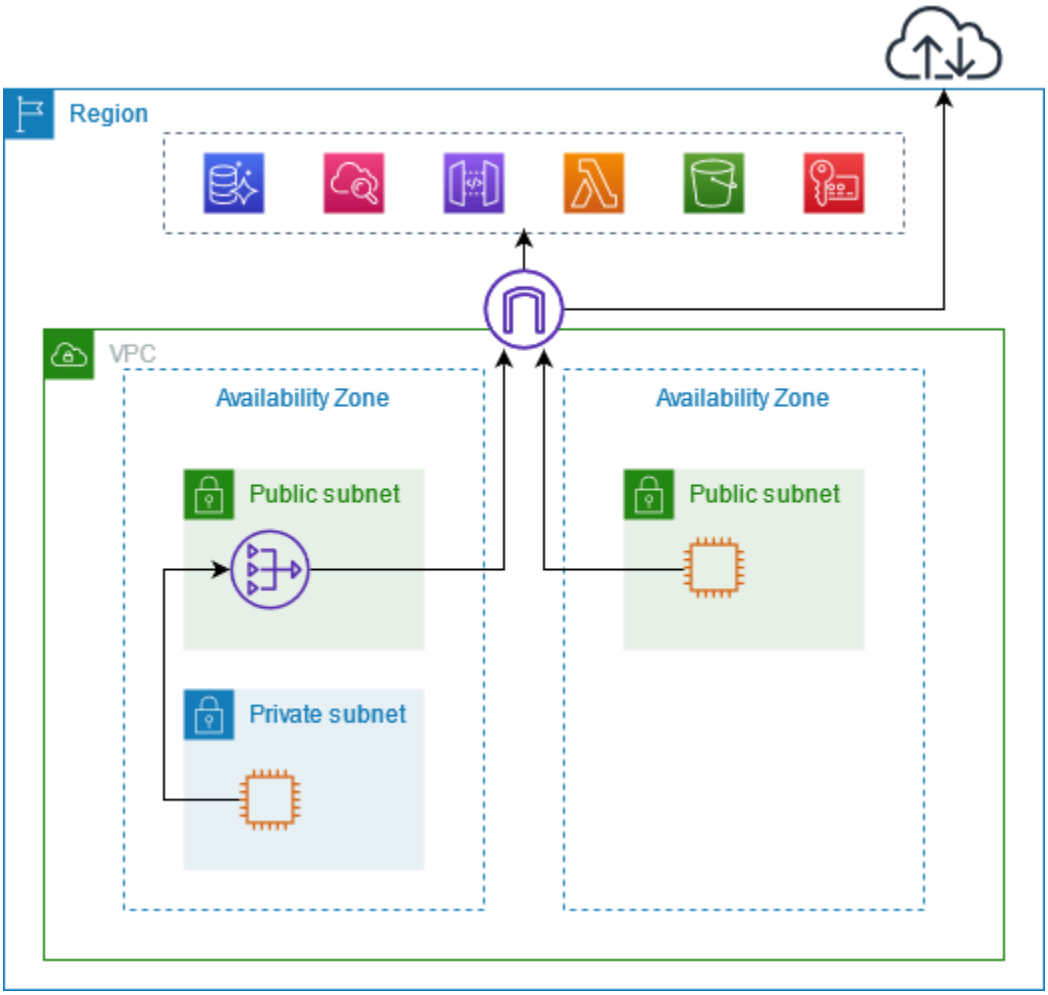
- Overview
- DNS hostnames
- DNS resolution
- Private DNS
- Subnets and Availability Zones
- IP address types
- DNS record IP type
- AWS services that integrate with AWS PrivateLink
- Cross-region enabled AWS services
- Access an AWS service using an interface VPC endpoint
- Configure an interface endpoint
- 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.

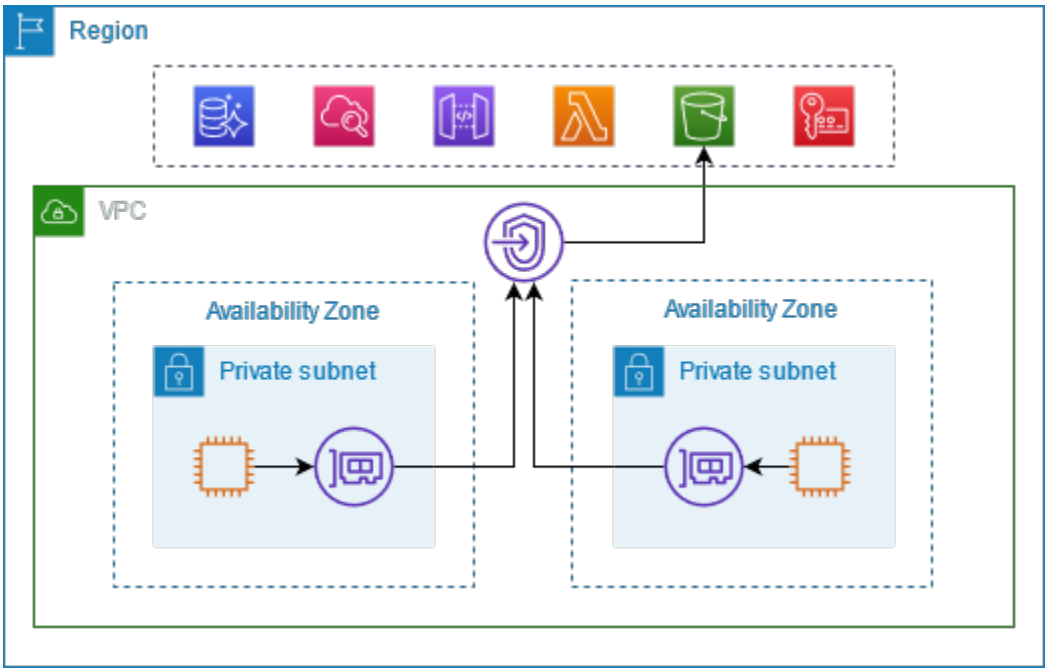
Access through public service endpoints

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-lj2wisx3-us-east-2b.monitoring.us-
east-2.vpce.amazonaws.com",
            "HostedZoneId": "ZC8PG0KIFKBRI"
        },
        {
            "DnsName": "monitoring.us-east-2.amazonaws.com",
            "HostedZoneId": "Z06320943MMOWYG6MAVL9"
        }
    ]
]
```

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.

We recommend that you enable private DNS names for your VPC endpoints for AWS services. This ensures that requests that use the public service endpoints, such as requests made through an AWS SDK, resolve to your VPC endpoint.

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.

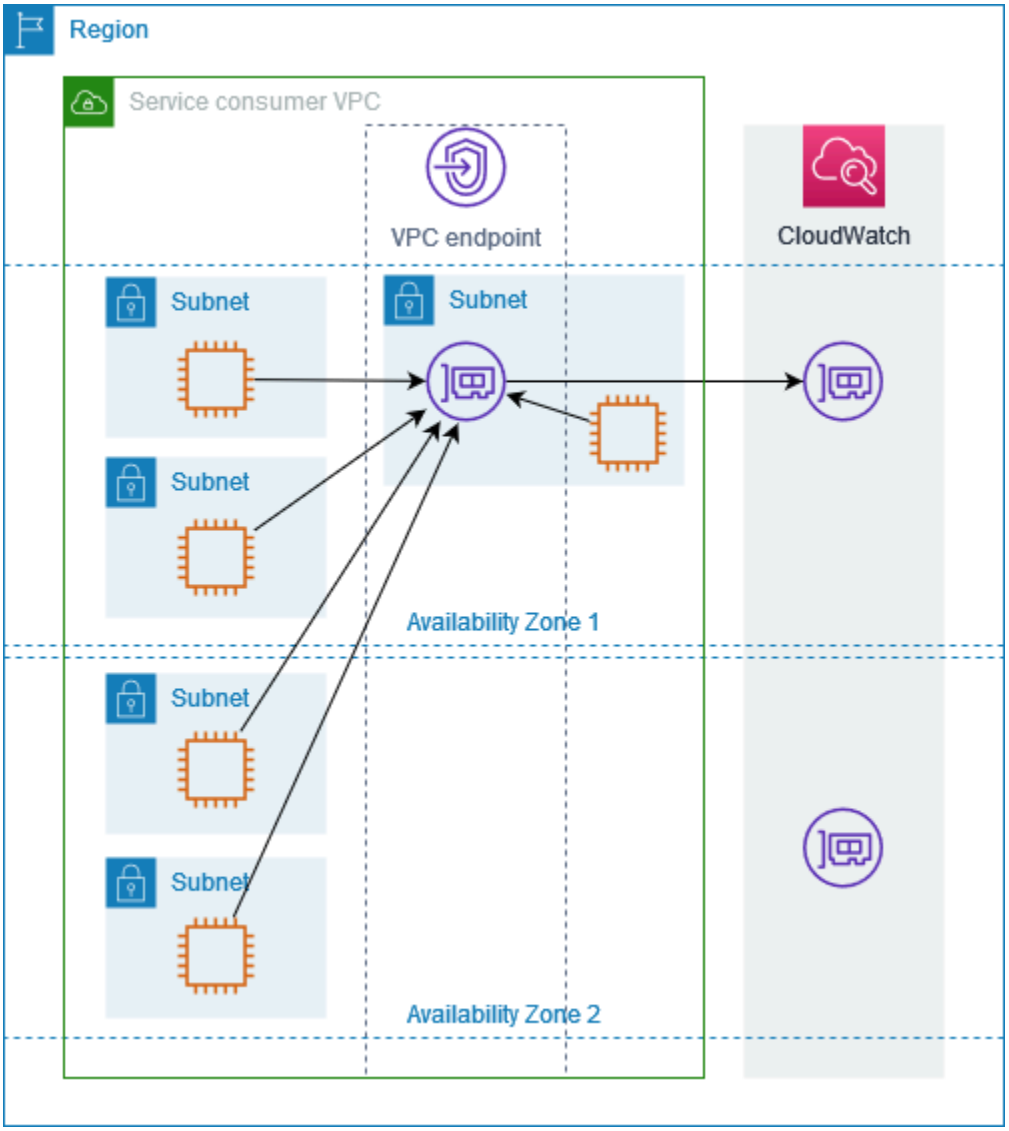
Subnets and Availability Zones

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.

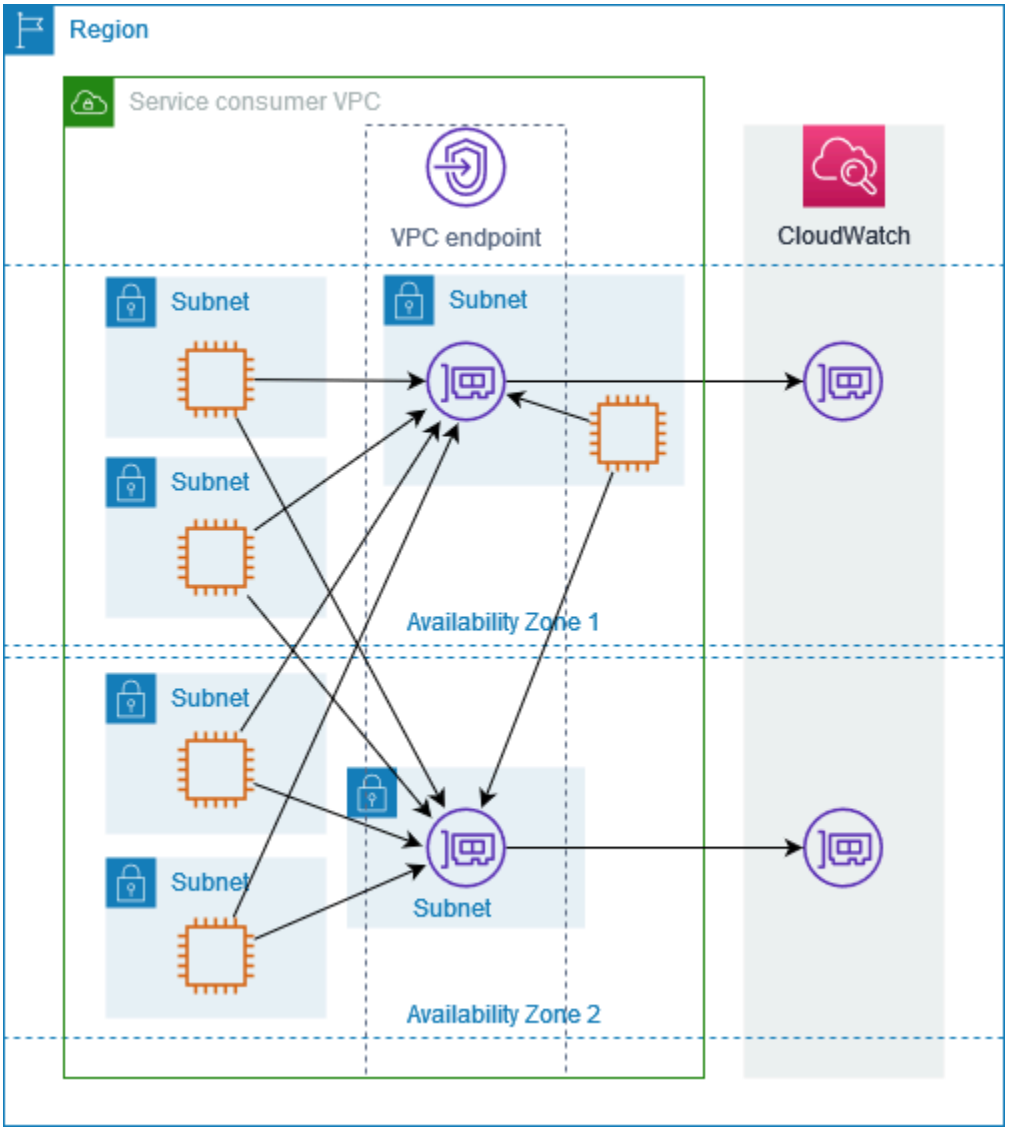
In a production environment, for high availability and resiliency, we recommend the following:

- 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.

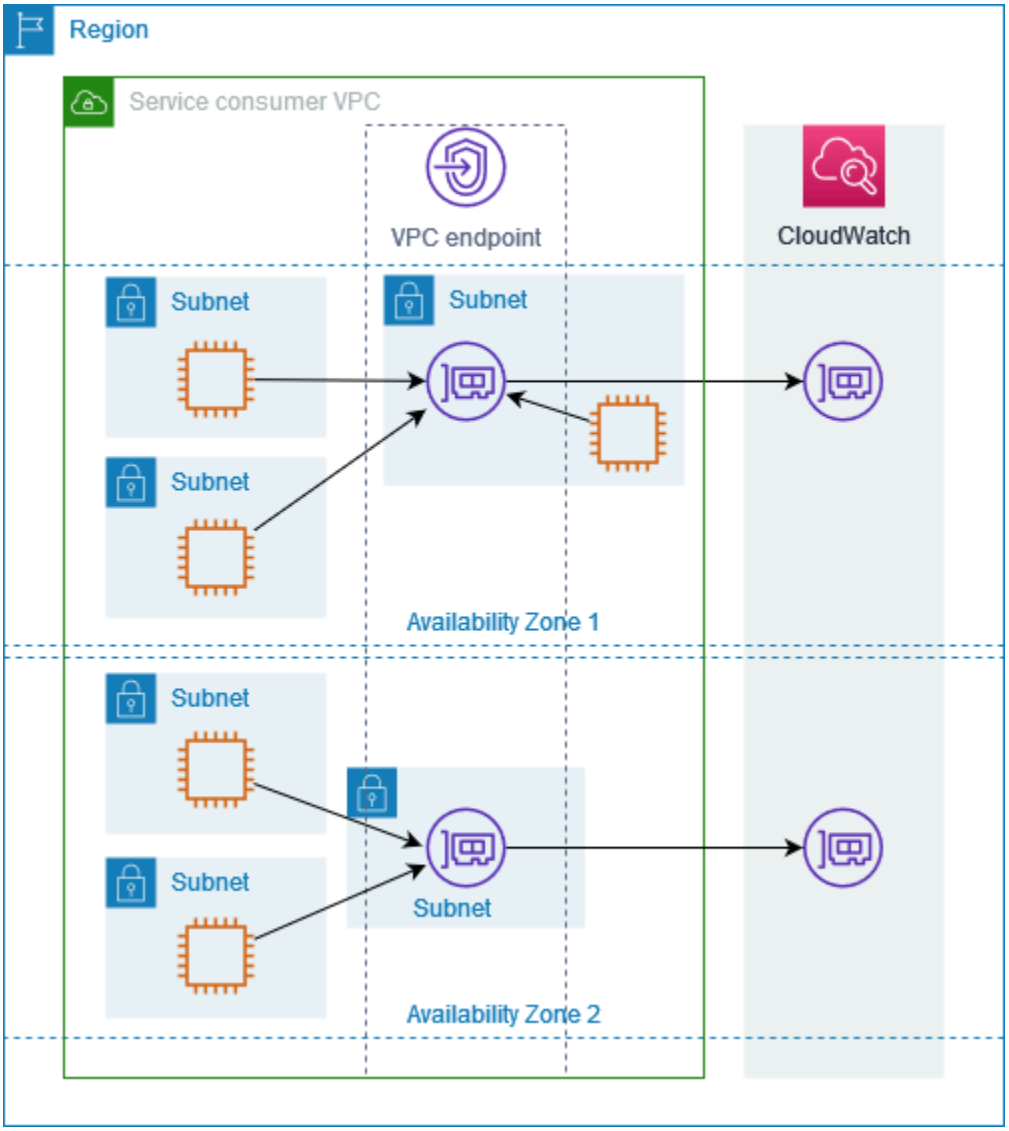
The following diagram shows a VPC endpoint for Amazon CloudWatch with an endpoint network interface in a single Availability Zone. When any resource in any subnet in the VPC accesses Amazon CloudWatch using its public endpoint, we resolve the traffic to the IP address of the endpoint network interface. 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.

DNS record IP type

Depending on your IP address type, when you call a VPC endpoint, the AWS service can return A records, AAAA records, or both A and AAAA records. You can customize which record types your AWS service returns by modifying the DNS record IP type. The following table shows the supported DNS record IP types and the returned record types:

DNS record IP type	Returned record types
IPv4	A
IPv6	AAAA
Dualstack	A and AAAA

By default, the DNS record type is the same as the IP address type. You can choose a different DNS record IP type, but you must use a compatible IP address type for the endpoint service. The following table shows the supported DNS record IP type for each IP address types for interface endpoints:

IP address type	Supported DNS record IP types
IPv4	IPv4

IP address type	Supported DNS record IP types
IPv6	IPv6
Dualstack	Dualstack*, IPv4, IPv6, service-defined

* Represents the default DNS record IP type.

A service-defined DNS record IP type returns DNS records based on the service endpoint you call. If you use a service-defined DNS record IP type, make sure your service can handle variable calls from service endpoints. To see the DNS records supported by your interface endpoint, see the DNS names for your VPC endpoint in the AWS Management Console, or use DescribeVpcEndpoints.

The DNS record IP type behavior is different for gateway endpoints. For more information, see DNS record IP type for gateway endpoints.

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 it indicates that the service manages the endpoint.

AWS service	Service name
AWS Account Management	com.amazonaws.*region*.account
Amazon API Gateway	com.amazonaws.*region*.execute-api
	com.amazonaws.*region*.apigateway
AWS AppConfig	com.amazonaws.*region*.appconfig
	com.amazonaws.*region*.appconfig-fips
	com.amazonaws.*region*.appconfigdata

AWS service	Service name
	com.amazonaws.*region*.appconfigdata-fips
AWS App Mesh	com.amazonaws.*region*.appmesh
	com.amazonaws.*region*.appmesh-envoy-management
AWS App Runner	com.amazonaws.*region*.apprunner
AWS App Runner services	com.amazonaws.*region*.apprunner.requests
Application Auto Scaling	com.amazonaws.*region*.application-autoscaling
AWS Application Discovery Service	com.amazonaws.*region*.discovery
	com.amazonaws.*region*.arsenal-discovery
AWS Application Migration Service	com.amazonaws.*region*.mgn
Amazon WorkSpaces Applications	com.amazonaws.*region*.appstream.api
	com.amazonaws.*region*.appstream.streaming
AWS AppSync	com.amazonaws.*region*.appsync-api
Amazon Athena	com.amazonaws.*region*.athena
AWS Audit Manager	com.amazonaws.*region*.auditmanager
Amazon Aurora	com.amazonaws.*region*.rds
	com.amazonaws.*region*.rds-fips
Amazon Aurora DSQL	com.amazonaws.*region*.dsql
AWS Auto Scaling	com.amazonaws.*region*.autoscaling-plans
AWS B2B Data Interchange	com.amazonaws.*region*.b2bi
AWS Backup	com.amazonaws.*region*.backup

AWS service	Service name
	com.amazonaws.*region*.backup-gateway
AWS Batch	com.amazonaws.*region*.batch
Amazon Bedrock	com.amazonaws.*region*.bedrock
	com.amazonaws.*region*.bedrock-agent
	com.amazonaws.*region*.bedrock-agent-runtime
	com.amazonaws.*region*.bedrock-data-automation
	com.amazonaws.*region*.bedrock-data-automation-fips
	com.amazonaws.*region*.bedrock-data-automation-runtime
	com.amazonaws.*region*.bedrock-data-automation-runtime-fips
	com.amazonaws.*region*.bedrock-runtime
Amazon Bedrock AgentCore	com.amazonaws.*region*.bedrock-agentcore-control
	com.amazonaws.*region*.bedrock-agentcore
AWS Billing and Cost Management	com.amazonaws.*region*.billing
	com.amazonaws.*region*.freetier
	com.amazonaws.*region*.tax
AWS Billing Conductor	com.amazonaws.*region*.billingconductor
Amazon Braket	com.amazonaws.*region*.braket
AWS Certificate Manager	com.amazonaws.*region*.acm
	com.amazonaws.*region*.acm-fips

AWS service	Service name
Amazon Chime SDK	com.amazonaws.*region*.chime-sdk-voice
	com.amazonaws.*region*.meetings-chime
	com.amazonaws.*region*.messaging-chime
	com.amazonaws.*region*.messaging-chime-fips
	com.amazonaws.*region*.identity-chime
	com.amazonaws.*region*.identity-chime-fips
	com.amazonaws.*region*.media-pipelines-chime
	com.amazonaws.*region*.media-pipelines-chime-fips
AWS Clean Rooms	com.amazonaws.*region*.cleanrooms
	com.amazonaws.*region*.cleanrooms-fips
AWS Clean Rooms ML	com.amazonaws.*region*.cleanrooms-ml
AWS Cloud Control API	com.amazonaws.*region*.cloudcontrolapi
	com.amazonaws.*region*.cloudcontrolapi-fips
Amazon Cloud Directory	com.amazonaws.*region*.clouddirectory
AWS CloudFormation	com.amazonaws.*region*.cloudformation
	com.amazonaws.*region*.cloudformation-fips
Amazon CloudFront	com.amazonaws.cloudfront
Amazon CloudFront Flat-Rate Pricing Plans	com.amazonaws.us-east-1.pricingplanmanager
AWS CloudHSM	com.amazonaws.*region*.cloudhsmv2
AWS Cloud Map	com.amazonaws.*region*.servicediscovery

AWS service	Service name
	com.amazonaws.*region*.servicediscovery-fips
	com.amazonaws.*region*.data-servicediscovery
	com.amazonaws.*region*.data-servicediscovery-fips
AWS CloudTrail	com.amazonaws.*region*.cloudtrail
AWS Cloud WAN	com.amazonaws.*region*.networkmanager
Amazon CloudWatch	com.amazonaws.*region*.application-signals
	com.amazonaws.*region*.applicationinsights
	com.amazonaws.*region*.internetmonitor
	com.amazonaws.*region*.internetmonitor-fips
	com.amazonaws.*region*.monitoring
	com.amazonaws.*region*.networkflowmonitor
	com.amazonaws.*region*.networkflowmonitorreports
	com.amazonaws.*region*.networkmonitor
	com.amazonaws.*region*.observabilityadmin
	com.amazonaws.*region*.rum
	com.amazonaws.*region*.rum-dataplane
	com.amazonaws.*region*.synthetics
	com.amazonaws.*region*.synthetics-fips
	com.amazonaws.*region*.oam
Amazon CloudWatch Logs	com.amazonaws.*region*.logs

AWS service	Service name
AWS CodeArtifact	com.amazonaws.*region*.codeartifact.api
	com.amazonaws.*region*.codeartifact.repositories
AWS CodeBuild	com.amazonaws.*region*.codebuild
	com.amazonaws.*region*.codebuild-fips
AWS CodeCommit	com.amazonaws.*region*.codecommit
	com.amazonaws.*region*.codecommit-fips
	com.amazonaws.*region*.git-codecommit
	com.amazonaws.*region*.git-codecommit-fips
AWS CodeConnections	com.amazonaws.*region*.codeconnections.api
	com.amazonaws.*region*.codestar-connections.api
AWS CodeDeploy	com.amazonaws.*region*.codedeploy
	com.amazonaws.*region*.codedeploy-commands-secure
	com.amazonaws.*region*.codedeploy-fips
Amazon CodeGuru Profiler	com.amazonaws.*region*.codeguru-profiler
Amazon CodeGuru Reviewer	com.amazonaws.*region*.codeguru-reviewer
AWS CodePipeline	com.amazonaws.*region*.codepipeline
Amazon Comprehend	com.amazonaws.*region*.comprehend
	com.amazonaws.*region*.comprehend-fips
Amazon Comprehend Medical	com.amazonaws.*region*.comprehendmedical
AWS Compute Optimizer	com.amazonaws.*region*.compute-optimizer

AWS service	Service name
AWS Config	com.amazonaws.*region*.config
	com.amazonaws.*region*.config-fips
Connect Customer	com.amazonaws.*region*.app-integrations
	com.amazonaws.*region*.cases
	com.amazonaws.*region*.connect
	com.amazonaws.*region*.connect-fips
	com.amazonaws.*region*.connect-campaigns
	com.amazonaws.*region*.profile
	com.amazonaws.*region*.voiceid
	com.amazonaws.*region*.wisdom
AWS Connector Service	com.amazonaws.*region*.awsconnector
AWS Control Catalog	com.amazonaws.*region*.controlcatalog
AWS Cost Explorer	com.amazonaws.*region*.ce
AWS Cost Optimization Hub	com.amazonaws.*region*.cost-optimization-hub
AWS Control Tower	com.amazonaws.*region*.controltower
	com.amazonaws.*region*.controltower-fips
AWS Data Exchange	com.amazonaws.*region*.dataexchange
AWS Data Exports	aws.api.*region*.bcm-data-exports
	com.amazonaws.*region*.bcm-pricing-calculator
Amazon Data Firehose	com.amazonaws.*region*.kinesis-firehose

AWS service	Service name
Amazon Data Lifecycle Manager	com.amazonaws.*region*.dlm
	com.amazonaws.*region*.dlm-fips
AWS Database Migration Service	com.amazonaws.*region*.dms
	com.amazonaws.*region*.dms-fips
AWS DataSync	com.amazonaws.*region*.datasync
Amazon DataZone	com.amazonaws.*region*.datazone
	com.amazonaws.*region*.datazone-fips
AWS Deadline Cloud	com.amazonaws.*region*.deadline.management
	com.amazonaws.*region*.deadline.scheduling
Amazon Detective	com.amazonaws.*region*.detective
	com.amazonaws.*region*.detective-fips
Amazon DevOps Guru	com.amazonaws.*region*.devops-guru
AWS Direct Connect	com.amazonaws.*region*.directconnect
	com.amazonaws.*region*.directconnect-fips
AWS Directory Service	com.amazonaws.*region*.ds
	com.amazonaws.*region*.ds-data
	com.amazonaws.*region*.ds-data-fips
Amazon DocumentDB	com.amazonaws.*region*.rds
Amazon DynamoDB	com.amazonaws.*region*.dynamodb
	com.amazonaws.*region*.dynamodb-fips

AWS service	Service name
	com.amazonaws.*region*.dynamodb-streams
Amazon EBS direct APIs	com.amazonaws.*region*.ebs
	com.amazonaws.*region*.ebs-fips
Amazon EC2	com.amazonaws.*region*.ec2
	com.amazonaws.*region*.ec2-fips
Amazon EC2 Auto Scaling	com.amazonaws.*region*.autoscaling
	com.amazonaws.*region*.autoscaling-fips
EC2 Image Builder	com.amazonaws.*region*.imagebuilder
Amazon ECR	com.amazonaws.*region*.ecr.api
	com.amazonaws.*region*.ecr.dkr
Amazon ECS	com.amazonaws.*region*.ecs
	com.amazonaws.*region*.ecs-agent
	com.amazonaws.*region*.ecs-telemetry
Amazon EKS	com.amazonaws.*region*.eks
	com.amazonaws.*region*.eks-auth
	com.amazonaws.*region*.eks-fips
	com.amazonaws.*region*.eks-mcp
	com.amazonaws.*region*.eks-proxy
	com.amazonaws.*region*.oidc-eks
AWS Elastic Beanstalk	com.amazonaws.*region*.elasticbeanstalk

AWS service	Service name
	com.amazonaws.*region*.elasticbeanstalk-health
AWS Elastic Disaster Recovery	com.amazonaws.*region*.drs
Amazon Elastic File System	com.amazonaws.*region*.elasticfilesystem
	com.amazonaws.*region*.elasticfilesystem-fips
Elastic Load Balancing	com.amazonaws.*region*.elasticloadbalancing
Amazon Elastic VMware Service	com.amazonaws.*region*.evs
	com.amazonaws.*region*.evs-fips
Amazon ElastiCache	com.amazonaws.*region*.elasticache
	com.amazonaws.*region*.elasticache-fips
AWS Elemental MediaConnect	com.amazonaws.*region*.mediaconnect
AWS Elemental MediaConvert	com.amazonaws.*region*.mediaconvert
	com.amazonaws.*region*.mediaconvert-fips
Amazon EMR	com.amazonaws.*region*.elasticmapreduce
	com.amazonaws.*region*.elasticmapreduce-fips
	com.amazonaws.*region*.emr-dashboard
Amazon EMR on EKS	com.amazonaws.*region*.emr-containers
Amazon EMR Serverless	com.amazonaws.*region*.emr-serverless
	com.amazonaws.*region*.emr-serverless-services.livy
	com.amazonaws.*region*.emr-serverless.dashboard
Amazon EMR WAL	com.amazonaws.*region*.emrwal.prod

AWS service	Service name
AWS End User Messaging Social	com.amazonaws.*region*.social-messaging
	com.amazonaws.*region*.social-messaging-fips
AWS Entity Resolution	com.amazonaws.*region*.entityresolution
	com.amazonaws.*region*.entityresolution-fips
Amazon EventBridge	com.amazonaws.*region*.events
	com.amazonaws.*region*.events-fips
	com.amazonaws.*region*.pipes
	com.amazonaws.*region*.pipes-data
	com.amazonaws.*region*.pipes-fips
	com.amazonaws.*region*.schemas
Amazon EventBridge Scheduler	com.amazonaws.*region*.scheduler
AWS Fault Injection Service	com.amazonaws.*region*.fis
	com.amazonaws.*region*.fis-fips
Amazon FinSpace	com.amazonaws.*region*.finspace
	com.amazonaws.*region*.finspace-api
AWS Firewall Manager	com.amazonaws.*region*.fms
	com.amazonaws.*region*.fms-fips
Amazon Forecast	com.amazonaws.*region*.forecast
	com.amazonaws.*region*.forecastquery
	com.amazonaws.*region*.forecast-fips

AWS service	Service name
	com.amazonaws.*region*.forecastquery-fips
Amazon Fraud Detector	com.amazonaws.*region*.frauddetector
Amazon FSx	com.amazonaws.*region*.fsx
	com.amazonaws.*region*.fsx-fips
Amazon GameLift Servers	com.amazonaws.*region*.gamelift
Amazon GameLift Streams	com.amazonaws.*region*.gameliftstreams
AWS Global Networks for Transit Gateways	com.amazonaws.*region*.networkmanager
AWS Glue	com.amazonaws.*region*.glue
	com.amazonaws.*region*.glue.dashboard
AWS Glue DataBrew	com.amazonaws.*region*.databrew
	com.amazonaws.*region*.databrew-fips
Amazon Managed Grafana	com.amazonaws.*region*.grafana
	com.amazonaws.*region*.grafana-workspace
AWS Ground Station	com.amazonaws.*region*.groundstation
	com.amazonaws.*region*.groundstation-fips
Amazon GuardDuty	com.amazonaws.*region*.guardduty
	com.amazonaws.*region*.guardduty-data
	com.amazonaws.*region*.guardduty-data-fips
	com.amazonaws.*region*.guardduty-fips
AWS HealthImaging	com.amazonaws.*region*.dicom-medical-imaging

AWS service	Service name
	com.amazonaws.*region*.medical-imaging
	com.amazonaws.*region*.runtime-medical-imaging
AWS HealthLake	com.amazonaws.*region*.healthlake
AWS HealthOmics	com.amazonaws.*region*.analytics-omics
	com.amazonaws.*region*.analytics-omics-fips
	com.amazonaws.*region*.control-storage-omics
	com.amazonaws.*region*.control-storage-omics-fips
	com.amazonaws.*region*.storage-omics
	com.amazonaws.*region*.tags-omics
	com.amazonaws.*region*.tags-omics-fips
	com.amazonaws.*region*.workflows-omics
	com.amazonaws.*region*.workflows-omics-fips
AWS Identity and Access Management (IAM)	com.amazonaws.iam
IAM Access Analyzer	com.amazonaws.*region*.access-analyzer
	com.amazonaws.*region*.access-analyzer-fips
IAM Identity Center	com.amazonaws.*region*.identitystore
	com.amazonaws.*region*.sso
	com.amazonaws.*region*.sso-oauth
IAM Roles Anywhere	com.amazonaws.*region*.rolesanywhere
	com.amazonaws.*region*.rolesanywhere-fips

AWS service	Service name
Amazon Inspector	com.amazonaws.*region*.inspector2
	com.amazonaws.*region*.inspector2-fips
	com.amazonaws.*region*.inspector-scan
	com.amazonaws.*region*.inspector-scan-fips
Amazon Interactive Video Service	com.amazonaws.*region*.ivs.contribute
AWS IoT Core	com.amazonaws.*region*.iot.api
	com.amazonaws.*region*.iot-fips.api
	com.amazonaws.*region*.iot.data
	com.amazonaws.*region*.iot.credentials
AWS IoT Device Management secure tunneling	com.amazonaws.*region*.iot.tunneling.api
	com.amazonaws.*region*.iot-fips.tunneling.api
	com.amazonaws.*region*.iot.tunneling.data
	com.amazonaws.*region*.iot-fips.tunneling.data
AWS IoT Core Device Advisor	com.amazonaws.*region*.deviceadvisor.iot
Managed integrations for AWS IoT Device Management	com.amazonaws.*region*.iotmanagedintegrations.api
	com.amazonaws.*region*.iotmanagedintegrations-fips.api
AWS IoT Core for LoRaWAN	com.amazonaws.*region*.iotwireless.api
	com.amazonaws.*region*.lorawan.cups
	com.amazonaws.*region*.lorawan.lns
AWS IoT FleetWise	com.amazonaws.*region*.iotfleetwise

AWS service	Service name
AWS IoT Greengrass	com.amazonaws.*region*.greengrass
AWS IoT RoboRunner	com.amazonaws.*region*.iotroborunner
AWS IoT SiteWise	com.amazonaws.*region*.iotsitewise.api
	com.amazonaws.*region*.iotsitewise.data
AWS IoT TwinMaker	com.amazonaws.*region*.iottwinmaker.api
	com.amazonaws.*region*.iottwinmaker.data
Amazon Kendra	com.amazonaws.*region*.kendra
	aws.api.*region*.kendra-ranking
AWS Key Management Service	com.amazonaws.*region*.kms
	com.amazonaws.*region*.kms-fips
Amazon Keyspaces (for Apache Cassandra)	com.amazonaws.*region*.cassandra
	com.amazonaws.*region*.cassandra-fips
Amazon Kinesis Data Streams	com.amazonaws.*region*.kinesis-streams
	com.amazonaws.*region*.kinesis-streams-fips
AWS Lake Formation	com.amazonaws.*region*.lakeformation
AWS Lambda	com.amazonaws.*region*.lambda
	com.amazonaws.*region*.lambda-microvm
AWS Launch Wizard	com.amazonaws.*region*.launchwizard
Amazon Lex V2	com.amazonaws.*region*.models-v2-lex
	com.amazonaws.*region*.runtime-v2-lex

AWS service	Service name
AWS License Manager	com.amazonaws.*region*.license-manager
	com.amazonaws.*region*.license-manager-fips
	com.amazonaws.*region*.license-manager-linux-subscriptions
	com.amazonaws.*region*.license-manager-linux-subscriptions-fips
	com.amazonaws.*region*.license-manager-user-subscriptions
	com.amazonaws.*region*.license-manager-user-subscriptions-fips
Amazon Lightsail	com.amazonaws.*region*.lightsail
Amazon Location Service	com.amazonaws.*region*.geo.maps
	com.amazonaws.*region*.geo.places
	com.amazonaws.*region*.geo.routes
	com.amazonaws.*region*.geo.geofencing
	com.amazonaws.*region*.geo.tracking
	com.amazonaws.*region*.geo.metadata
Amazon Lookout for Equipment	com.amazonaws.*region*.lookoutequipment
Amazon Macie	com.amazonaws.*region*.macie2
	com.amazonaws.*region*.macie2-fips
AWS Mainframe Modernization	com.amazonaws.*region*.apptest
	com.amazonaws.*region*.m2

AWS service	Service name
Amazon Managed Blockchain	com.amazonaws.*region*.managedblockchain-query
	com.amazonaws.*region*.managedblockchain.bitcoin.mainnet
	com.amazonaws.*region*.managedblockchain.bitcoin.testnet
AWS Marketplace Agreement	com.amazonaws.*region*.agreement-marketplace
AWS Marketplace Discovery	com.amazonaws.*region*.discovery-marketplace
AWS Marketplace Metering Service	com.amazonaws.*region*.metering-marketplace
Amazon Managed Service for Prometheus	com.amazonaws.*region*.aps
	com.amazonaws.*region*.aps-workspaces
Amazon Managed Streaming for Apache Kafka (MSK)	com.amazonaws.*region*.kafka
	com.amazonaws.*region*.kafka-fips
Amazon Managed Workflows for Apache Airflow	com.amazonaws.*region*.airflow.api
	com.amazonaws.*region*.airflow.api-fips
	com.amazonaws.*region*.airflow.env
	com.amazonaws.*region*.airflow.env-fips
	com.amazonaws.*region*.airflow.ops
Amazon Route 53	com.amazonaws.route53
Amazon Route 53 Global Resolver	aws.api.us-east-2.route53globalresolver
	aws.api.us-east-2.route53globalresolver-fips
Amazon Route 53 Profiles	com.amazonaws.*region*.route53profiles

AWS service	Service name
AWS Management Console	com.amazonaws.*region*.console
	com.amazonaws.*region*.signin
Amazon MemoryDB	com.amazonaws.*region*.memory-db
	com.amazonaws.*region*.memorydb-fips
AWS Migration Hub Orchestrator	com.amazonaws.*region*.migrationhub-orchestrator
AWS Migration Hub Refactor Spaces	com.amazonaws.*region*.refactor-spaces
Migration Hub Strategy Recommendations	com.amazonaws.*region*.migrationhub-strategy
Amazon MQ	com.amazonaws.*region*.mq
	com.amazonaws.*region*.mq-fips
Amazon Neptune Analytics	com.amazonaws.*region*.neptune-graph
	com.amazonaws.*region*.neptune-graph-data
	com.amazonaws.*region*.neptune-graph-fips
AWS Network Firewall	com.amazonaws.*region*.network-firewall
	com.amazonaws.*region*.network-firewall-fips
Amazon OpenSearch Service	These endpoints are service-managed
Amazon OpenSearch Ingestion	com.amazonaws.*region*.osis
AWS Organizations	com.amazonaws.*region*.organizations
	com.amazonaws.*region*.organizations-fips
AWS Outposts	com.amazonaws.*region*.outposts

AWS service	Service name
AWS Payment Cryptography	com.amazonaws.*region*.payment-cryptography.controlplane
	com.amazonaws.*region*.payment-cryptography.dataplane
AWS PCS	com.amazonaws.*region*.pcs
	com.amazonaws.*region*.pcs-fips
Amazon Personalize	com.amazonaws.*region*.personalize
	com.amazonaws.*region*.personalize-events
	com.amazonaws.*region*.personalize-runtime
Amazon Pinpoint	com.amazonaws.*region*.pinpoint
	com.amazonaws.*region*.pinpoint-sms-voice-v2
Amazon Polly	com.amazonaws.*region*.polly
	com.amazonaws.*region*.polly-fips
AWS Price List	com.amazonaws.*region*.pricing.api
AWS Private Certificate Authority	com.amazonaws.*region*.acm-pca
	com.amazonaws.*region*.acm-pca-fips
	com.amazonaws.*region*.pca-connector-ad
	com.amazonaws.*region*.pca-connector-scep
AWS Proton	com.amazonaws.*region*.proton
Amazon Q Business	aws.api.*region*.qbusiness
Amazon Q Developer	com.amazonaws.*region*.codewhisperer

AWS service	Service name
	com.amazonaws.*region*.q
	com.amazonaws.*region*.qapps
Amazon Q User Subscriptions	com.amazonaws.*region*.service.user-subscriptions
Quick	com.amazonaws.*region*.quicksight-website
Amazon RDS	com.amazonaws.*region*.rds
	com.amazonaws.*region*.rds-fips
Amazon RDS Data API	com.amazonaws.*region*.rds-data
Amazon RDS Performance Insights	com.amazonaws.*region*.pi
	com.amazonaws.*region*.pi-fips
AWS re:Post Private	com.amazonaws.*region*.repostspace
Recycle Bin	com.amazonaws.*region*.rbin
	com.amazonaws.*region*.rbin-fips
Amazon Redshift	com.amazonaws.*region*.redshift
	com.amazonaws.*region*.redshift-fips
	com.amazonaws.*region*.redshift-serverless
	com.amazonaws.*region*.redshift-serverless-fips
Amazon Redshift Data API	com.amazonaws.*region*.redshift-data
	com.amazonaws.*region*.redshift-data-fips
Amazon Rekognition	com.amazonaws.*region*.rekognition
	com.amazonaws.*region*.rekognition-fips

AWS service	Service name
	com.amazonaws.*region*.streaming-rekognition
	com.amazonaws.*region*.streaming-rekognition-fips
AWS Resource Access Manager	com.amazonaws.*region*.ram
	com.amazonaws.*region*.ram-fips
AWS Resource Explorer	com.amazonaws.*region*.resource-explorer-2
	com.amazonaws.*region*.resource-explorer-2-fips
AWS Resource Groups	com.amazonaws.*region*.resource-groups
	com.amazonaws.*region*.resource-groups-fips
AWS Resource Groups Tagging API	com.amazonaws.*region*.tagging
Amazon S3	com.amazonaws.*region*.s3
	com.amazonaws.*region*.s3-fips
	com.amazonaws.*region*.s3tables
Amazon S3 Multi-Region Access Points	com.amazonaws.s3-global.accesspoint
Amazon S3 on Outposts	com.amazonaws.*region*.s3-outposts
Amazon SageMaker AI	aws.sagemaker.*region*.experiments
	aws.sagemaker.*region*.notebook
	aws.sagemaker.*region*.partner-app
	aws.sagemaker.*region*.studio
	com.amazonaws.*region*.sagemaker-data-science-assistant

AWS service	Service name
	com.amazonaws.*region*.sagemaker.api
	com.amazonaws.*region*.sagemaker.api-fips
	com.amazonaws.*region*.sagemaker.featurestore-runtime
	com.amazonaws.*region*.sagemaker.featurestore-runtime-fips
	com.amazonaws.*region*.sagemaker.metrics
	com.amazonaws.*region*.sagemaker.runtime
	com.amazonaws.*region*.sagemaker.runtime-fips
Savings Plans	com.amazonaws.savingsplans
AWS Secrets Manager	com.amazonaws.*region*.secretsmanager
AWS Security Agent	com.amazonaws.*region*.securityagent
AWS Security Hub CSPM	com.amazonaws.*region*.securityhub
	com.amazonaws.*region*.securityhub-fips
Amazon Security Lake	com.amazonaws.*region*.securitylake
	com.amazonaws.*region*.securitylake-fips
AWS Security Token Service	com.amazonaws.*region*.sts
	com.amazonaws.*region*.sts-fips
AWS Serverless Application Repository	com.amazonaws.*region*.serverlessrepo
Service Catalog	com.amazonaws.*region*.servicecatalog

AWS service	Service name
	com.amazonaws.*region*.servicecatalog-appregistry
Service Quotas	com.amazonaws.*region*.servicequotas
Amazon SES	com.amazonaws.*region*.email-smtp
	com.amazonaws.*region*.mail-manager
	com.amazonaws.*region*.mail-manager-fips
	com.amazonaws.*region*.mail-manager-smtp.auth.fips
	com.amazonaws.*region*.mail-manager-smtp.open.fips
AWS Snowball Edge Device Management	com.amazonaws.*region*.snow-device-management
Amazon SNS	com.amazonaws.*region*.sns
Amazon SQS	com.amazonaws.*region*.sqs
	com.amazonaws.*region*.sqs-fips
Amazon SWF	com.amazonaws.*region*.swf
	com.amazonaws.*region*.swf-fips
AWS Step Functions	com.amazonaws.*region*.states
	com.amazonaws.*region*.sync-states
AWS Storage Gateway	com.amazonaws.*region*.storagegateway
AWS Supply Chain	com.amazonaws.*region*.scn
AWS Systems Manager	com.amazonaws.*region*.ec2messages
	com.amazonaws.*region*.ssm
	com.amazonaws.*region*.ssm-contacts

AWS service	Service name
	com.amazonaws.*region*.ssm-incidents
	com.amazonaws.*region*.ssm-incidents-fips
	com.amazonaws.*region*.ssm-quicksetup
	com.amazonaws.*region*.ssmmessages
AWS Systems Manager for SAP	com.amazonaws.*region*.ssm-sap
	com.amazonaws.*region*.ssm-sap-fips
AWS Telco Network Builder	com.amazonaws.*region*.tnb
Amazon Textract	com.amazonaws.*region*.textract
	com.amazonaws.*region*.textract-fips
Amazon Timestream	com.amazonaws.*region*.timestream.ingest-*cell*
	com.amazonaws.*region*.timestream.query-*cell*
Amazon Timestream for InfluxDB	com.amazonaws.*region*.timestream-influxdb
	com.amazonaws.*region*.timestream-influxdb-fips
Amazon Transcribe	com.amazonaws.*region*.transcribe
	com.amazonaws.*region*.transcribestreaming
com.amazonaws.*region*.transcribestreaming-fips	
Amazon Transcribe Medical	com.amazonaws.*region*.transcribe
	com.amazonaws.*region*.transcribestreaming
AWS Transfer Family	com.amazonaws.*region*.transfer

AWS service	Service name
AWS Transform	com.amazonaws.*region*.transform
AWS Transform custom	com.amazonaws.*region*.transform-custom
Amazon Translate	com.amazonaws.*region*.translate
AWS Trusted Advisor	com.amazonaws.*region*.trustedadvisor
AWS User Notifications	com.amazonaws.*region*.notifications
	com.amazonaws.*region*.notifications-contacts
Amazon Verified Permissions	com.amazonaws.*region*.verifiedpermissions
	com.amazonaws.*region*.verifiedpermissions-fips
Amazon VPC Lattice	com.amazonaws.*region*.vpc-lattice
AWS WAFV2	com.amazonaws.*region*.wafv2
	com.amazonaws.*region*.wafv2-fips
AWS Well-Architected Tool	com.amazonaws.*region*.wellarchitected
Amazon WorkMail	com.amazonaws.*region*.workmail
	com.amazonaws.*region*.workmailmessageflow
Amazon WorkSpaces	com.amazonaws.*region*.workspaces
Amazon WorkSpaces Secure Browser	com.amazonaws.*region*.workspaces-web
	com.amazonaws.*region*.workspaces-web-fips
WorkSpaces streaming	com.amazonaws.*region*.highlander
Amazon WorkSpaces Thin Client	com.amazonaws.*region*.thinclient.api
aws.api.*region*.s3files	

AWS service	Service name
aws.api.*region*.s3files-fips	
AWS X-Ray	com.amazonaws.*region*.xray
Amazon Managed Service for Apache Flink	com.amazonaws.*region*.kinesisanalytics
	com.amazonaws.*region*.kinesisanalytics-fips

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. The complete output is not shown.

```
[
    "api.aws.us-east-1.cassandra-streams",
    "aws.api.us-east-1.bcm-data-exports",
    "aws.api.us-east-1.emr-service-cell01",
    "aws.api.us-east-1.freetier",
    "aws.api.us-east-1.kendra-ranking",
    "aws.api.us-east-1.qbusiness",
    . . .
     "com.amazonaws.us-east-1.xray"
]
```

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"
                },
                {
                    "PrivateDnsName": "monitoring.us-east-1.api.aws"
                },
                {
```

```
                    "PrivateDnsName": "monitoring-fips.us-east-1.amazonaws.com"
                },
                {
                    "PrivateDnsName": "monitoring-fips.us-east-1.api.aws"
                }           ],
            "VpcEndpointPolicySupported": true,
            "AcceptanceRequired": false,
            "ManagesVpcEndpoints": false,
            "Tags": [],
            "PrivateDnsNameVerificationState": "verified",
            "SupportedIpAddressTypes": [
                "ipv6",
                "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. The complete output is not shown.

```
[
    "api.aws.us-east-1.cassandra-streams",
    "aws.api.us-east-1.bcm-data-exports",
    "aws.api.us-east-1.emr-service-cell01",
    "aws.api.us-east-1.freetier",
    "aws.api.us-east-1.kendra-ranking",
    . . .
    "com.amazonaws.us-east-1.xray"
]
```

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. The complete output is not shown.

```
[
    "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",
    . . .
```

```
]
```

View IPv6 support

To view IPv6 support for AWS services, see AWS services that support IPv6. You can also 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. The complete output is not shown.

```
[
    "api.aws.us-east-1.cassandra-streams",
    "aws.api.us-east-1.bcm-data-exports",
    "aws.api.us-east-1.freetier",
    "aws.api.us-east-1.kendra-ranking",
    "aws.api.us-east-1.qbusiness",
    "aws.api.us-east-1.resource-explorer-2",
    "aws.api.us-east-1.resource-explorer-2-fips",
    "aws.sagemaker.us-east-1.experiments",
    "aws.sagemaker.us-east-1.partner-app",
    "com.amazonaws.iam",
    "com.amazonaws.us-east-1.access-analyzer",
    "com.amazonaws.us-east-1.account",
    . . .
    "com.amazonaws.us-east-1.xray"
]
```

Cross-region enabled AWS services

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

Choose the link in the **AWS service** column to see the service documentation. The **Service name** column contains the service name that you specify when you create the interface endpoint.

AWS service	Service name
Amazon S3	com.amazonaws.*region*.s3
AWS Identity and Access Management (IAM)	com.amazonaws.iam
Amazon ECR	com.amazonaws.*region*.ecr.api
	com.amazonaws.*region*.ecr.dkr
AWS Key Management Service	com.amazonaws.*region*.kms
	com.amazonaws.*region*.kms-fips
Amazon ECS	com.amazonaws.*region*.ecs
AWS Lambda	com.amazonaws.*region*.lambda
Amazon Data Firehose	com.amazonaws.*region*.kinesis-firehose
Amazon Managed Service for Apache Flink	com.amazonaws.*region*.kinesisanalytics
	com.amazonaws.*region*.kinesisanalytics-fips
Amazon Route 53	com.amazonaws.route53

View available AWS service names

You can use the describe-vpc-endpoint-services command to view cross Region enabled services.

The following example displays the AWS services that a user in the `us-east-1` Region can access over interface endpoints, to the specified (`us-west-2`) service 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 \
  --service-region us-west-2 \
  --query ServiceNames
```

The following is example output. The complete output is not shown.

```
[
    "com.amazonaws.us-west-2.ecr.api",
    "com.amazonaws.us-west-2.ecr.dkr",
    "com.amazonaws.us-west-2.ecs",
    "com.amazonaws.us-west-2.ecs-fips",
    ...
    "com.amazonaws.us-west-2.s3"
]
```

Note

You must use regional DNS. Zonal DNS is not supported when accessing AWS services in another Region. For more information, see View and update DNS attributes in the Amazon VPC User Guide.

Permissions and Considerations

- By default, IAM entities don't have permission to access an AWS service in another Region. To grant the permissions required for cross Region access, an IAM administrator can create IAM policies that allow the `vpce:AllowMultiRegion` permission-only action.
- Ensure that your Service Control Policy (SCP) does not deny `vpce:AllowMultiRegion` permission-only action. To use AWS PrivateLink's cross-region connectivity feature, both your identity policy and your SCP must allow this action.
- To control the Regions that an IAM entity can specify as a service Region when creating a VPC endpoint, use the `ec2:VpceServiceRegion` condition key.
- A service consumer must opt in to an opt-in Region before selecting it as the service Region for an endpoint. Whenever possible, we recommend that service consumers access a service using intra-Region connectivity instead of cross-Region connectivity. Intra-Region connectivity provides lower latency and lower costs.
- You can use IAM's new `aws:SourceVpcArn` global condition key to secure which Regions, AWS accounts and VPCs your resources can be accessed from. This key helps implement data residency and region based access control.
- For high availability, create a cross Region enabled interface endpoint in at least two Availability Zones. In this case, providers and consumers are not required to use the same Availability Zones.

- With cross Region access, AWS PrivateLink manages failover between Availability Zones in both service and consumer Regions. It does not manage failover across Regions.
- Cross Region access is not supported for the following Availability Zones: `use1-az3`, `usw1-az2`, `apne1-az3`, `apne2-az2`, and `apne2-az4`.
- You can use AWS Fault Injection Service to simulate regional events and model failure scenarios for in-region and cross-region enabled interface endpoints. To learn more, see AWS FIS documentation.

Create an interface endpoint to an AWS service in another Region

To create an interface endpoint using the Console, see the Create a VPC endpoint section.

In the CLI, you can use the create-vpc-endpoint command to create a VPC endpoint to an AWS service in a different Region. The following example creates an interface endpoint to Amazon S3 in `us-west-2` from a VPC in `us-east-1`.

```
aws ec2 create-vpc-endpoint \
  --vpc-id vpc-id \
  --service-name com.amazonaws.us-west-2.s3 \
  --vpc-endpoint-type Interface \
  --subnet-ids subnet-id-1 subnet-id-2 \
  --region us-east-1 \
  --service-region us-west-2
```

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
- ICMP

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 for the endpoint network interface that allows the expected traffic from the resources in your VPC. For example, to ensure that the AWS CLI can send HTTPS requests 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 resources in your VPC and the endpoint network interfaces.
- 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 **Type**, choose **AWS services**.
5. (Optional) If creating an endpoint to an AWS service in another Region, select the **Enable cross Region endpoint** checkbox and then select the **service region** from the drop down.

6. For **Service name**, select the service. For more information, see the section called "Services that integrate".
7. For **VPC**, select the VPC from which you'll access the AWS service.
8. 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. This ensures that requests that use the public service endpoints, such as requests made through an AWS SDK, resolve to your VPC endpoint.
9. For **Subnets**, select the subnets in which to create endpoint network interfaces. You can select one subnet per Availability Zone. You can't select multiple subnets from the same Availability Zone. For more information, see the section called "Subnets and Availability Zones".

   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 yourself, select **Designate IP addresses**. Note that the first four IP addresses and the last IP address in a subnet CIDR block are reserved for internal use, so you can't specify them for your endpoint network interfaces.
10. For **IP address type**, choose from the following options:

    - **IPv4** – Assign IPv4 addresses to the 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 the 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 the 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.
11. For **Security groups**, select the security groups to associate with the endpoint network interfaces. By default, we associate the default security group for the VPC.

12. For **Policy**, to allow all operations by all principals on all resources over the interface endpoint, select **Full access**. To restrict access, select **Custom** and enter a policy. This option is available only if the service supports VPC endpoint policies. For more information, see Endpoint policies.
13. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
14. 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. VPC endpoints that are managed by an AWS service (service managed VPC endpoints) can be created by the service in a shared subnet.

ICMP

Interface endpoints do not respond to **ping** requests. You can use the **nc** or **nmap** commands instead.

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 processes 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

We recommend that you enable private DNS names for your VPC endpoints for AWS services. This ensures that requests that use the public service endpoints, such as requests made through an AWS SDK, resolve to 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 Amazon Resource Name (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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "vpce.amazonaws.com"
            },
            "Action": "SNS:Publish",
            "Resource": "arn:aws:sns:us-east-1:111111111111:topic-name",
            "Condition": {
```

```
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ec2:us-east-1:111111111111:vpc-
endpoint/endpoint-id"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111111111111"
                }
            }
        }
    ]
}
```

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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "vpce.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111111111111:key/key-id",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ec2:us-east-1:111111111111:vpc-
endpoint/endpoint-id"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111111111111"
                }
            }
```

```
        }
    ]
}
```

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

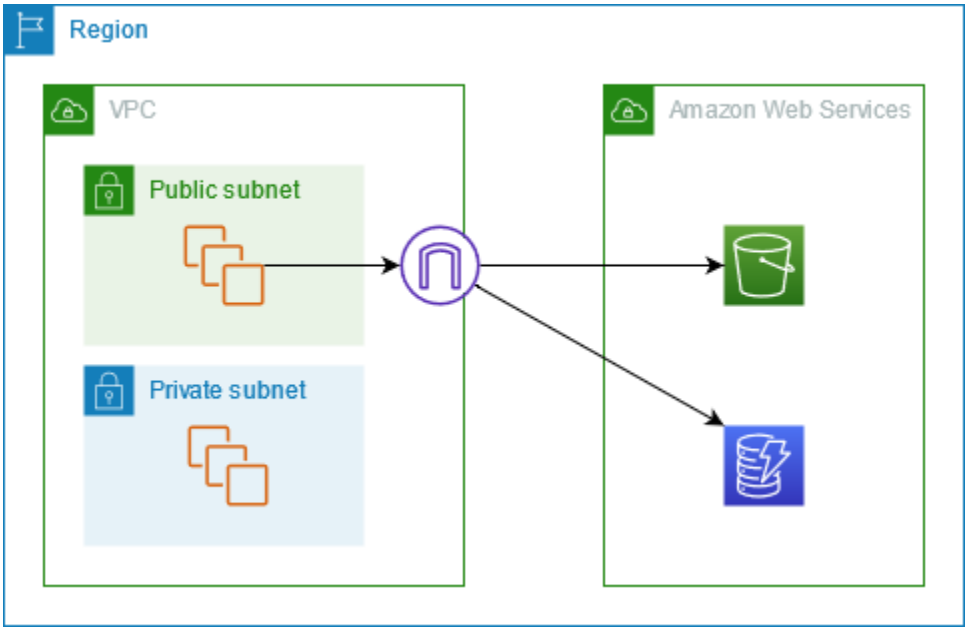
- Overview
- Routing
- Security
- IP address type
- DNS record IP type
- Gateway endpoints for Amazon S3
- Gateway endpoints for Amazon DynamoDB

Overview

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

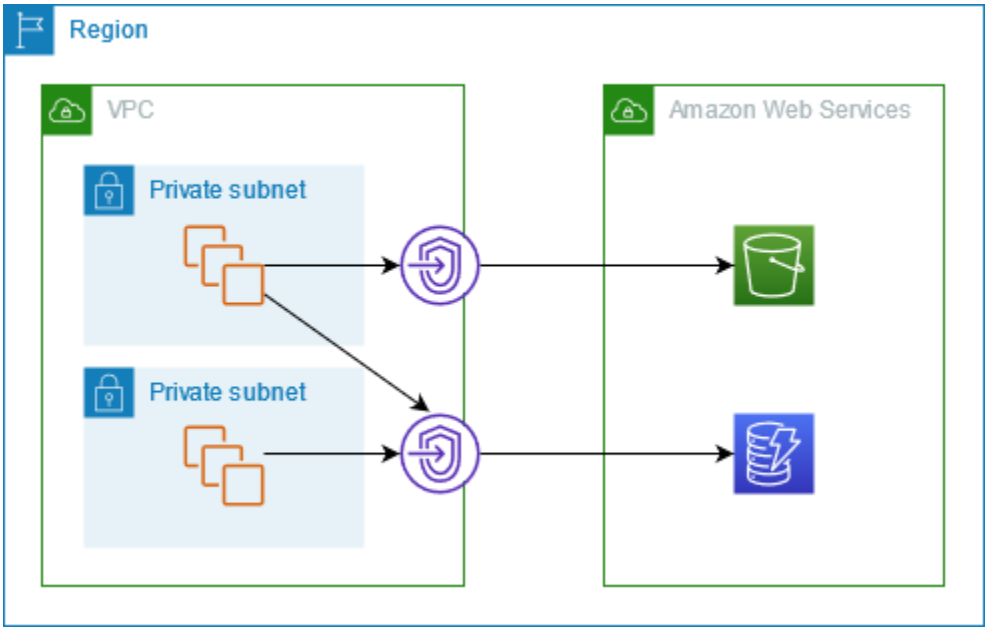
Access through an internet gateway

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
prefix_list_id	*gateway_endpoint_id*

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
prefix_list_id	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
service_cidr_block_1	TCP	443
service_cidr_block_2	TCP	443
service_cidr_block_3	TCP	443

IP address type

The IP address type determines which prefix list is associated with your route table.

Requirements to enable IPv6 for a gateway endpoint

- The IP address type of a gateway endpoint must be compatible with the subnets for the gateway endpoint, as described here:
  - **IPv4** – Add the service's IPv4 prefix list to your route table.
  - **IPv6** – Add the service's IPv6 prefix list to your route table. This option is supported only if all selected subnets are IPv6 only subnets.

- **Dualstack** – Add the service's IPv4 prefix list to your route table and add the service's IPv6 prefix list to your route table. This option is supported only if all selected subnets have both IPv4 and IPv6 address ranges.

DNS record IP type

By default, a gateway endpoint returns DNS records based on the service endpoint you call. If you create your gateway endpoint using the IPv4 service endpoint, such as `s3.us-east-2.amazonaws.com`, Amazon S3 returns A records to your clients, and all subnets in your route table use IPv4.

In contrast, if you create your gateway endpoint using the dualstack service endpoint, such as `s3.dualstack.us-east-2.amazonaws.com`, Amazon S3 returns both A and AAAA records to your clients, and the subnets in your route table use IPv4 and IPv6.

> **Note**
>
> For directory buckets, or S3 Express One Zone, the gateway endpoints for the data plane would be `s3express-use2-az1.us-east-2.amazonaws.com` and `s3express-use2-az1.dualstack.us-east-2.amazonaws.com` respectively.

The DNS record IP type affects how traffic is routed to your clients. If you create a gateway endpoint using the IPv4 service endpoint and then call the dualstack service endpoint, traffic that uses AAAA records won't be routed through the gateway endpoint. Traffic will be dropped or routed over an IPv6-compatible path if one is present. If you use a service-defined DNS record IP type, make sure your service can handle variable calls from multiple service endpoints.

Instead of the default DNS record IP type setting of service-defined, you can customize the DNS record IP type to choose which records are returned for a specific endpoint. The following table shows the supported DNS record IP types and the returned record types:

DNS record IP type	Returned record types
IPv4	A
IPv6	AAAA

DNS record IP type	Returned record types
Dualstack	A and AAAA
service-defined	The records depend on the service endpoint

To choose a DNS record IP type, you must use a compatible IP address type for the endpoint service. The following table shows the supported DNS record IP type for each IP address types for gateway endpoints:

IP address type	Supported DNS record IP types
IPv4	IPv4, service-defined*
IPv6	IPv6, service-defined*
Dualstack	IPv4, IPv6, Dualstack, service-defined*

* Represents the default DNS record IP type.

> **Note**
>
> To use DNS record IP types other than service-defined for your Gateway endpoint, you must allow `enableDnsSupport` and `enableDnsHostnames` attributes in your VPC settings.

You can't change the DNS record IP type for a DynamoDB gateway endpoint. DynamoDB only supports the DNS record IP type of service-defined.

The DNS record IP type behavior is different for interface endpoints. For more information, see DNS record IP type for interface endpoints.

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.
- The source IPv4 or IPv6 addresses from instances in your affected subnets as received by Amazon S3 change from public addresses to the private addresses in your VPC. An endpoint switches network routes, and disconnects open TCP connections. The previous connections that used public 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 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

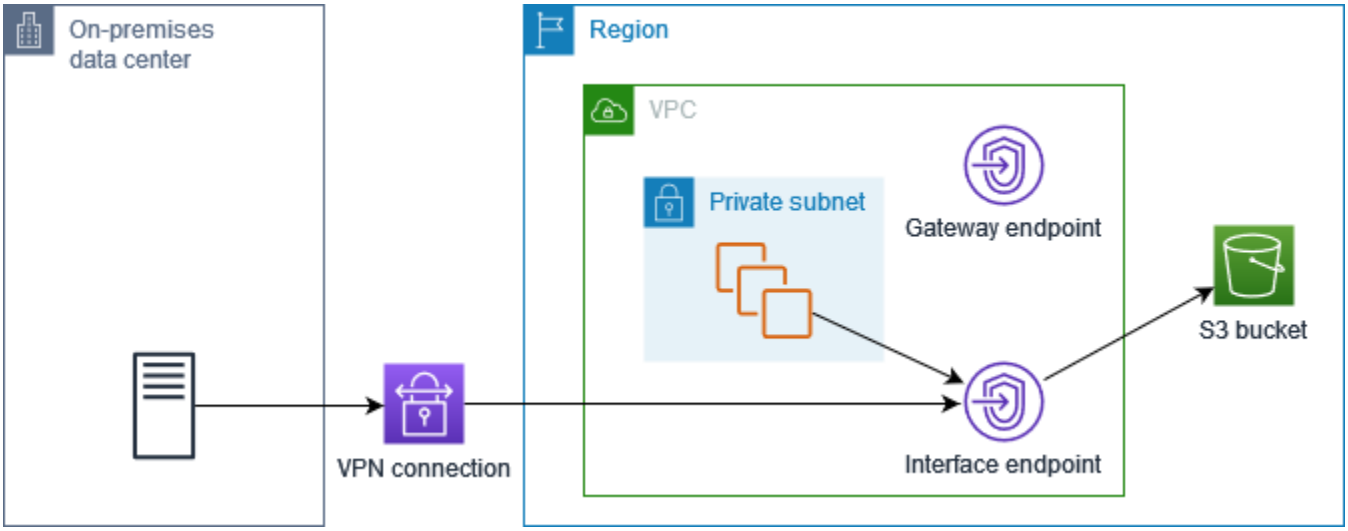
Amazon provides a DNS server, called the Route 53 Resolver, for your VPC. The Route 53 Resolver automatically resolves local VPC domain names and records in private hosted zones. However, you can't use the Route 53 Resolver from outside your VPC. Route 53 provides Resolver endpoints and Resolver rules 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

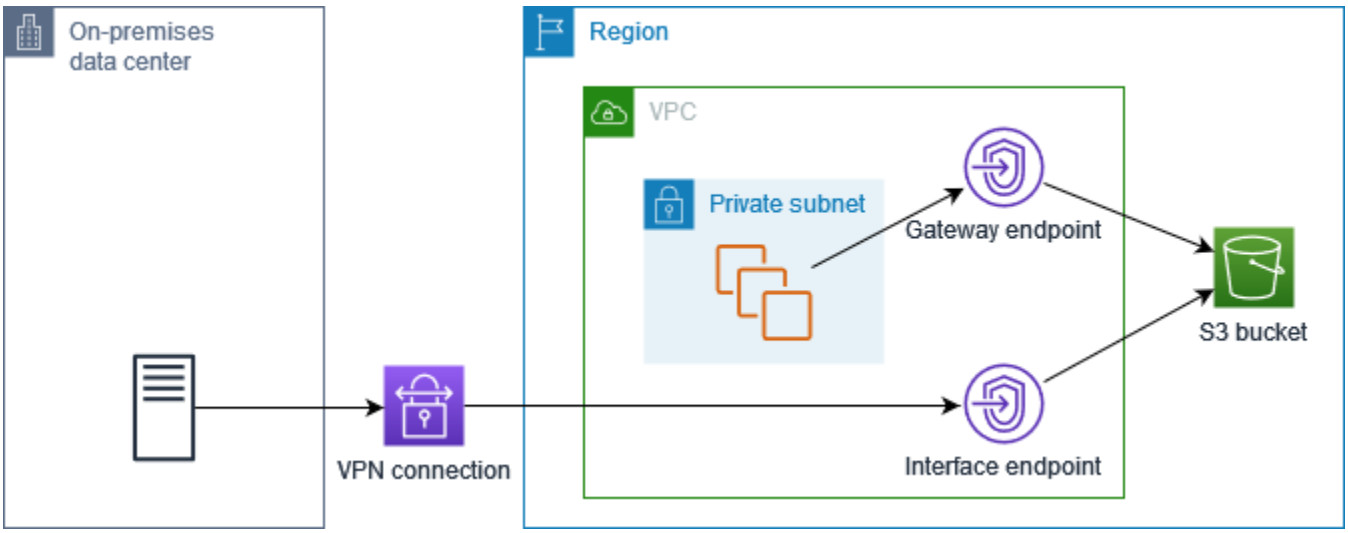
If you configure private DNS for your interface endpoint for Amazon S3 but do not configure private DNS only for the inbound Resolver endpoint, requests from both your on-premises network and your VPC use the interface endpoint to access Amazon S3. Therefore, you pay to use the interface endpoint for 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.

In order to configure this, the DNS record IP type of the gateway endpoint must match the interface endpoint or be `service-defined`. AWS PrivateLink doesn't support any other combination. For more information, see the section called “DNS record IP type”.



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**.

   If your Amazon S3 data is stored in general purpose buckets, select **com.amazonaws.*region*.s3**.

   If your Amazon S3 data is stored in directory buckets, select **com.amazonaws.*region*.s3express**.
6. For **VPC**, select the VPC in which to create the endpoint.
7. For **IP address type**, choose from the following options:

- **IPv4** – Assign IPv4 addresses to the 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 the 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 the 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.

8. 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.
9. 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.
10. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
11. 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: 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.

JSON

```
{
  "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: 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.

JSON

```
{
  "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: 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.

JSON

```
{
  "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: 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.

JSON

```
{
  "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. After you update an endpoint policy, it can take a few minutes for the changes to take effect. 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: 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.

JSON

```
{
```

```
  "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: 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.

JSON

```
{
  "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: Restrict access to users in a specific account

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

JSON

```
{
  "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 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: 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.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "Allow-access-from-specific-endpoint",
         "Effect": "Deny",
         "Principal": "*",
```

```
            "Action": "dynamodb:*",
            "Resource": "arn:aws:dynamodb:us-east-1:111111111111:table/*",
            "Condition": {
               "StringNotEquals" : {
                  "aws:sourceVpce": "vpce-11aa22bb"
               }
            }
         }
   ]
}
```

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.

JSON

```
{
   "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: 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.

JSON

```
{
   "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. After you update an endpoint policy, it can take a few minutes for the changes to take effect. 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: 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: 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

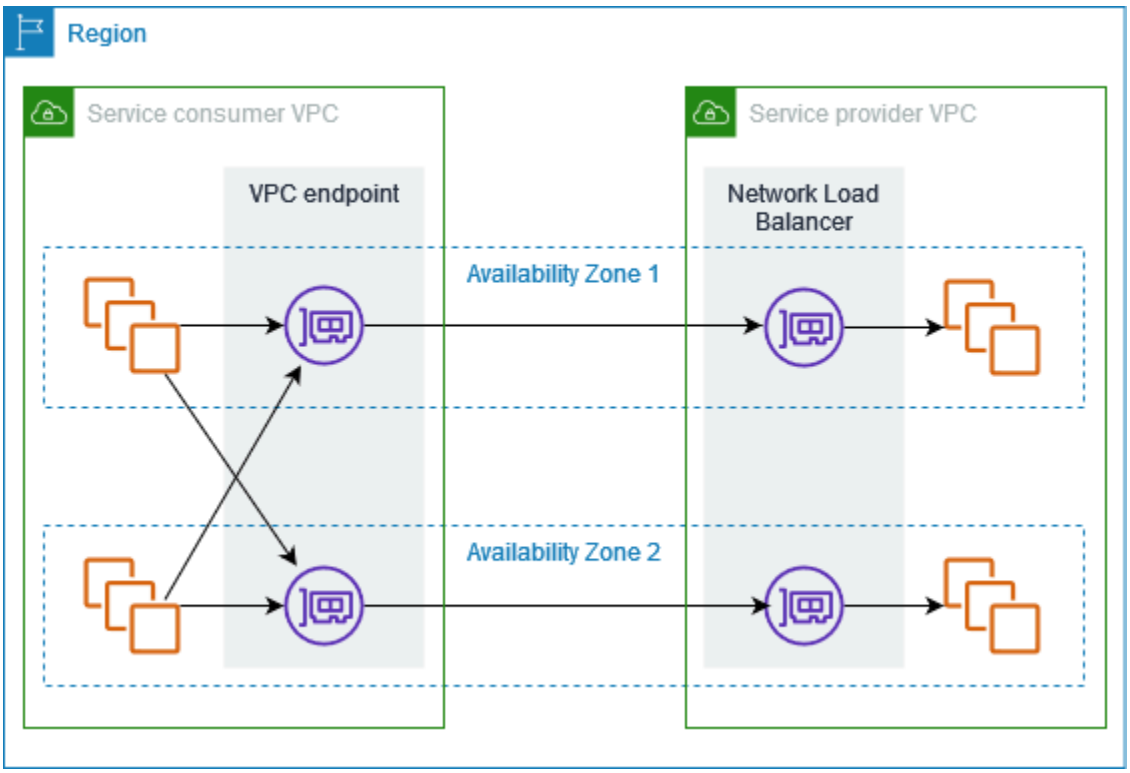
- Overview
- Create an interface endpoint

Overview

You can discover, purchase, and provision SaaS products powered by AWS PrivateLink through AWS Marketplace. For more information, see Access SaaS applications securely and privately using AWS 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 **Type**, choose **AWS Marketplace services**.
    b. Select the service.
5. If you subscribed to a service with the AWS Service Ready designation, do the following:

   a. For **Type**, choose **PrivateLink Ready partner services**.
   b. Enter the name of the service, and then choose **Verify service**.
6. For **VPC**, select the VPC from which you'll access the product.
7. For **Subnets**, select the subnets in which to create endpoint network interfaces.
8. For **Security groups**, 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”.

Share your services through AWS PrivateLink

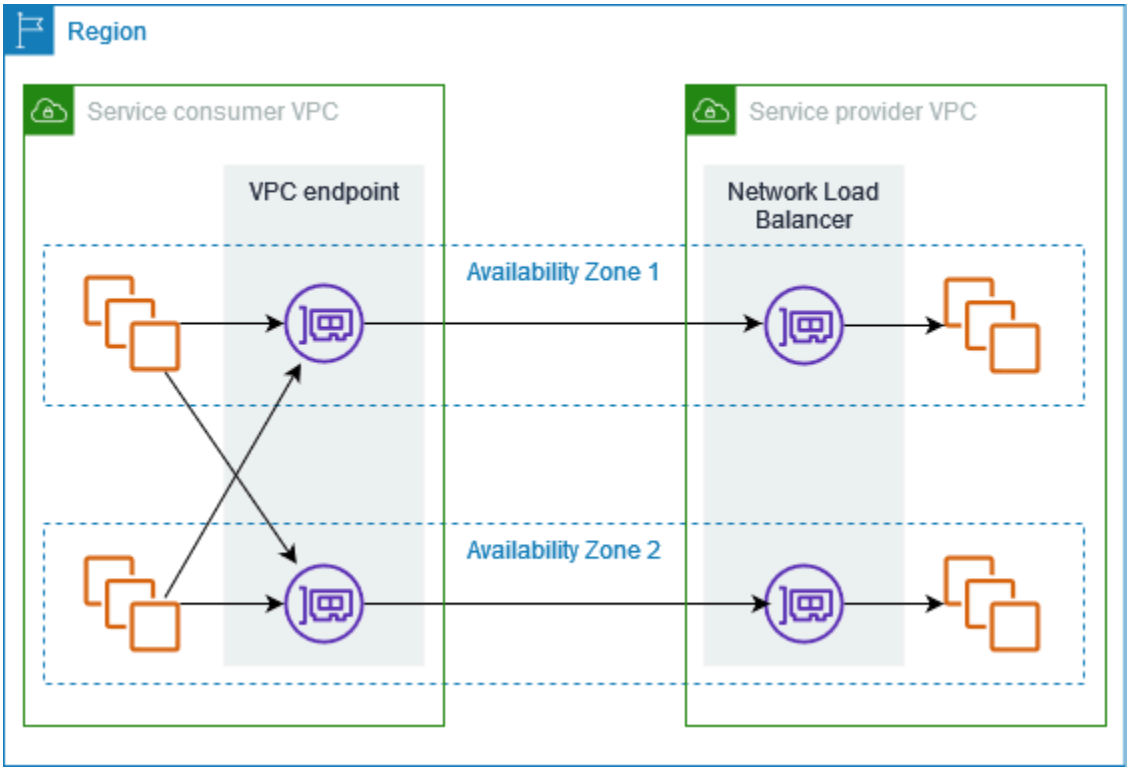
You can host your own AWS PrivateLink powered service, known as an *endpoint service*, and share it with other AWS customers.

Contents

- Overview
- DNS hostnames
- Private DNS
- Subnets and Availability Zones
- Cross-Region access
- 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.service_region.vpce.amazonaws.com
```

Zonal names have the following syntax:

```
endpoint_id-endpoint_zone.endpoint_service_id.service_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.

Subnets and Availability Zones

Your endpoint service is available in the Availability Zones that you enable for your Network Load Balancer. For high availability and resiliency, we recommend that you enable your load balancer in at least two Availability Zones, deploy EC2 instances in each enabled zone, and register these instances with your load balancer target group.

You can enable cross-zone load balancing as an alternative to hosting your endpoint service in multiple Availability Zones. However, consumers will lose access to the endpoint service from both zones if the zone that hosts the endpoint service fails. Also consider that when you enable cross-zone load balancing for a Network Load Balancer, EC2 data transfer charges apply.

The consumer can create interface VPC endpoints in the Availability Zones in which your endpoint service is available. We create an endpoint network interface in each subnet that the consumer configures for the VPC endpoint. We assign IP addresses to each endpoint network interface from its subnet, based on the IP address type of the VPC endpoint. When a request uses the regional endpoint for the VPC endpoint service, we select a healthy endpoint network interface, using the round robin algorithm to alternate between the network interfaces in different Availability Zones. We then resolve the traffic to the IP address of the selected endpoint network interface.

The consumer can use the zonal endpoints for the VPC endpoint if it's better for their use case to keep traffic in the same Availability Zone.

Cross-Region access

A service provider can host a service in one Region and make it available in a set of supported Regions. A service consumer selects a service Region when creating an endpoint.

Permissions

- By default, IAM entities don't have permission to make an endpoint service available in multiple Regions or access an endpoint service across Regions. To grant the permissions required for cross-Region access, an IAM administrator can create IAM policies that allow the `vpce:AllowMultiRegion` permission-only action.
- To control the Regions that an IAM entity can specify as a supported Region when creating an endpoint service, use the `ec2:VpceSupportedRegion` condition key.
- To control the Regions that an IAM entity can specify as a service Region when creating a VPC endpoint, use the `ec2:VpceServiceRegion` condition key.

Considerations

- A service provider must opt in to an opt-in Region before adding it as a supported Region for an endpoint service.
- Your endpoint service must be accessible from its host Region. You can't remove the host Region from the set of supported Regions. For redundancy, you can deploy your endpoint service in multiple Regions and enable cross-Region access for each endpoint service.
- A service consumer must opt in to an opt-in Region before selecting it as the service Region for an endpoint. Whenever possible, we recommend that service consumers access a service using intra-Region connectivity instead of cross-Region connectivity. Intra-Region connectivity provides lower latency and lower costs.
- If a service provider removes a Region from the set of supported Regions, service consumers can't select that Region as the service Region when they create new endpoints. Note that this does not affect access to the endpoint service from existing endpoints that use this Region as the service Region.
- For high availability, providers must use at least two Availability Zones. Cross-Region access does not require that providers and consumers use the same Availability Zones.
- Cross-Region access is not supported for the following Availability Zones: `use1-az3`, `usw1-az2`, `apne1-az3`, `apne2-az2`, and `apne2-az4`.
- With cross-Region access, AWS PrivateLink manages failover between Availability Zones. It does not manage failover across Regions.
- Cross-Region access is not supported for Network Load Balancers with a custom value configured for the TCP idle timeout.
- Cross-Region access is not supported with UDP fragmentation.

- Cross-Region access is only supported for services that you share through AWS PrivateLink.

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
- Connect to an endpoint service as the service consumer

Considerations

- An endpoint service is available in the Region where you created it. Consumers can access your service from other Regions if you enable cross-Region access, or if they use VPC peering or a transit gateway.

- 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*.
- 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*.
- A Network Load Balancer can be associated with a single endpoint service, but an endpoint service can be associated with multiple 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.

- If your Network Load Balancer has a security group, it must allow inbound traffic from the IP addresses of the clients. Alternatively, you can turn off evaluation of inbound security group rules for traffic through AWS PrivateLink. For more information, see Security groups in the *User Guide for Network Load Balancers*.
- 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. To see the Availability Zones that are enabled for the load balancer you selected, see **Details of selected load balancers**, **Included Availability Zones**. Your endpoint service will be available in these Availability Zones.
6. (Optional) To make your endpoint service available from Regions other than the Region where it is hosted, select the Regions from **Service Regions**. For more information, see the section called “Cross-Region access”.

7. 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.
8. 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.
9. 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.
10. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
11. 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 section called “Connect to an endpoint service as the service consumer”.
- 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 **Type**, choose **Endpoint services that use NLBs and GWLBs**.
5. For **Service name**, enter the name of the service (for example, `com.amazonaws.vpce.us-east-1.vpce-svc-0e123abc123198abc`), and then choose **Verify service**.
6. (Optional) To connect to an endpoint service that is available in a Region other than the endpoint Region, select **Service Region**, **Enable Cross Region endpoint**, and then select the Region. For more information, see the section called “Cross-Region access”.
7. For **VPC**, select the VPC from which you'll access the endpoint service.
8. For **Subnets**, select the subnets in which to create endpoint network interfaces.
9. For **IP address type**, choose from the following options:

   - **IPv4** – Assign IPv4 addresses to the 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 the 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 the 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.
10. 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**.

11. For **Security group**, select the security groups to associate with the endpoint network interfaces.
12. 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
- Manage load balancers
- Associate a private DNS name
- Modify the supported Regions
- 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

*

Considerations

- 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.
- If you remove permissions, it does not affect existing connections between the endpoint and the service that were previously accepted.

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.

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.

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”.

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)

Manage load balancers

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

If you enable another Availability Zone for your load balancers, the Availability Zone will appear under the **Load Balancers** tab on the **Endpoint services** page. However, it won't be enabled for the endpoint service or listed in the **Details** tab of your endpoint service on the AWS Management Console. You need to enable the endpoint service for the new Availability Zone.

It might take a few minutes for the load balancer's Availability Zone to be ready for your endpoint service. If you are using an automation, we recommend that you add a wait in your automation process before you enable the endpoint service for the new Availability Zone.

To manage 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. Change the endpoint service configuration as needed. For example:
   - Select the check box for a load balancer to associate it with the endpoint service.
   - Clear the check box for a load balancer to disassociate it from the endpoint service. You must keep at least one load balancer selected.
6. Choose **Save changes**

   The endpoint service will be enabled for any new Availability Zones you added to your load balancer. The new Availability Zone is listed under the **Load Balancers** tab and the **Details** tab of the endpoint service.

After you enable an Availability Zone for the endpoint service, service consumers can add a subnet from that Availability Zone to their interface VPC endpoints.

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

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

To enable the endpoint service in an Availability Zone that was recently enabled for the load balancer, simply call the command with the ID of the endpoint service.

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 Regions

You can modify the set of supported Regions for your endpoint service. Before you can add an opt-in Region, you must opt in. You can't remove the Region that hosts your endpoint service.

After you remove a Region, service consumers can't create new endpoints that specify it as the service Region. Removing a Region doesn't affect existing endpoints that specify it as the service Region. When you remove a Region, we recommend that you reject any existing endpoint connections from that Region.

To modify the supported Regions for your endpoint service

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 supported Regions**.
5. Select and deselect Regions as needed.
6. Choose **Save changes**.

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. Suppose that a service provider makes their service available through a public endpoint and as an endpoint service. If the service provider uses the DNS name of the public endpoint as the private DNS name of the endpoint service, then service consumers can access the public endpoint or the endpoint service using the same client application, without modification. If a request comes from the service consumer VPC, the private DNS servers resolve the DNS name to the IP addresses of the endpoint network interfaces. Otherwise, the public DNS servers resolve the DNS name to the public endpoint.

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.
- When the consumer creates an interface endpoint to connect to your service, we create a private hosted zone and associate it with the service consumer VPC. We create a CNAME record in the private hosted zone that maps the private DNS name of the endpoint service to the regional DNS name of the VPC endpoint. When a consumer sends a request to the public DNS name of the service, the private DNS servers resolve the request to the IP addresses of the endpoint network interfaces.
- 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*. 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.

- Private DNS names are not supported for Gateway Load Balancer endpoints.

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-vpc-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-vpc-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:l6p0ERxlTt45jevFwOCp",
        "Name": "_6e86v84tqgqubxbwii1m"
    }
]
```

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
_6e86v84tqgqubxbwii1m.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 "_6e86v84tqgqubxbwii1m" and simply use "example.com" in the TXT record.

After we verify "_6e86v84tqgqubxbwii1m.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 using a simple routing policy. Use the following values:

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

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** command, 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  _6e86v84tqgqubxbwii1m.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.

   ```
   _6e86v84tqgqubxbwii1m.example.com text = "vpce:l6p0ERxlTt45jevFwOCp"
   ```

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, "_6e86v84tqgqubxbwii1m") 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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "vpce.amazonaws.com"
            },
            "Action": "SNS:Publish",
            "Resource": "arn:aws:sns:us-east-1:111111111111:topic-name",
            "Condition": {
                "ArnLike": {
```

```
                    "aws:SourceArn": "arn:aws:ec2:us-east-1:111111111111:vpc-
endpoint-service/service-id"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111111111111"
                }
            }
        }
    ]
}
```

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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "vpce.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-east-1:111111111111:key/key-id",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ec2:us-east-1:111111111111:vpc-
endpoint-service/service-id"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111111111111"
                }
            }
        }
```

```
    ]
}
```

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)

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.

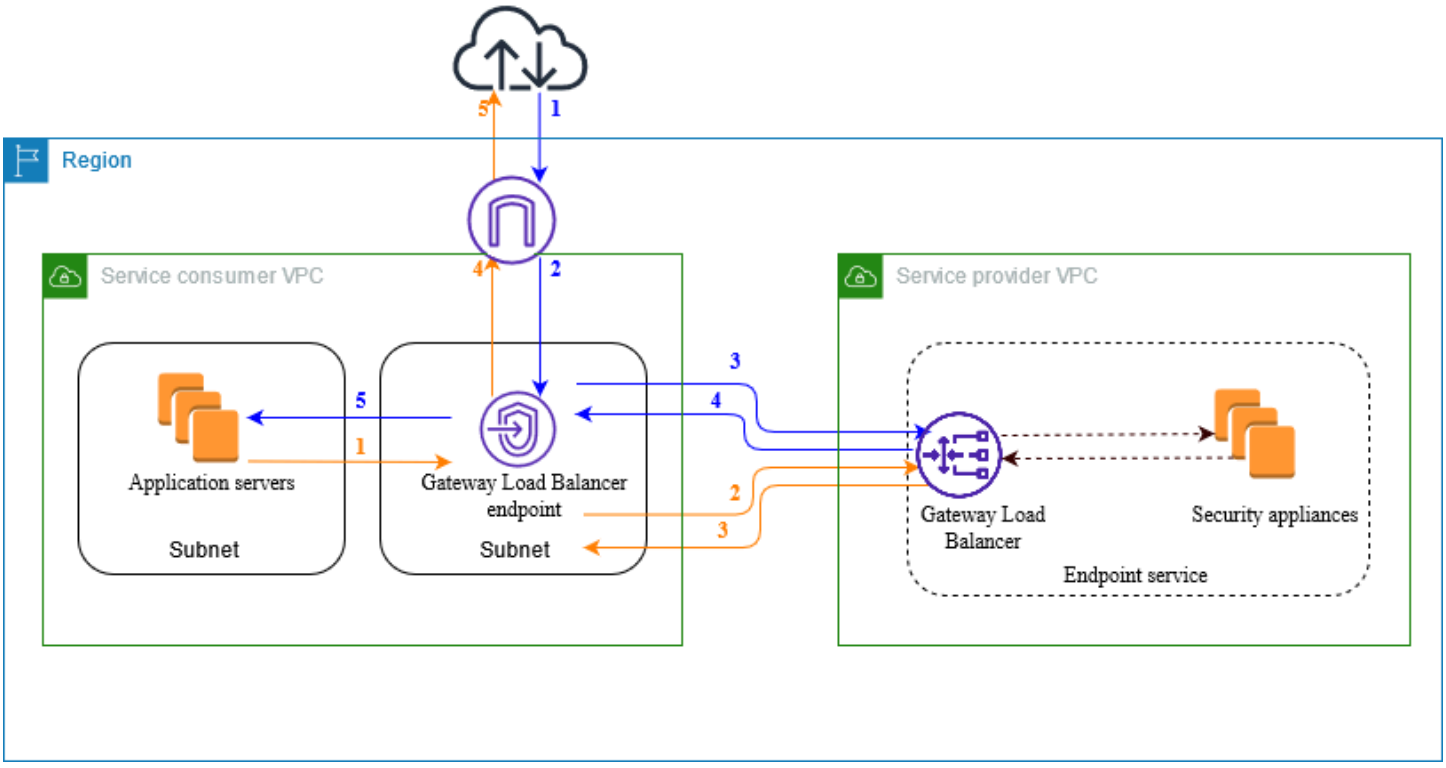
Contents

- 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. When using a Gateway Load Balancer endpoint as a target, you cannot specify a prefix list as a destination. In these tables, 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
VPC IPv4 CIDR	Local
VPC IPv6 CIDR	Local
Application subnet IPv4 CIDR	*vpc-endpoint-id*
Application subnet IPv6 CIDR	*vpc-endpoint-id*

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
VPC IPv4 CIDR	Local
VPC IPv6 CIDR	Local
0.0.0.0/0	*vpc-endpoint-id*
::/0	*vpc-endpoint-id*

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
VPC IPv4 CIDR	Local
VPC IPv6 CIDR	Local
0.0.0.0/0	*internet-gateway-id*
::/0	*internet-gateway-id*

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*.
- 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*.
- 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.
- If the application servers and the Gateway Load Balancer endpoint are in the same subnet, the NACL rules are evaluated for traffic from the application servers to the Gateway Load Balancer endpoint.
- If you use a Gateway Load Balancer with an egress-only internet gateway, the IPv6 traffic is dropped. Instead, use an internet gateway and inbound firewall rules.
- 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 **Type**, choose **Endpoint services that use NLBs and GWLBs**.
5. For **Service name**, enter the name of the service, and then choose **Verify service**.
6. For **VPC**, select the VPC from which you'll access the endpoint service.

7. For **Subnets**, select one subnet in which to create an endpoint network interface.
8. For **IP address type**, choose from the following options:
   - **IPv4** – Assign IPv4 addresses to the endpoint network interface. This option is supported only if the selected subnet has an IPv4 address range.
   - **IPv6** – Assign IPv6 addresses to the endpoint network interface. This option is supported only if the selected subnet is an IPv6 only subnet.
   - **Dualstack** – Assign both IPv4 and IPv6 addresses to the endpoint network interface. This option is supported only if the selected subnet has 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)

Access VPC resources through AWS PrivateLink

You can privately access a VPC resource in another VPC using a resource VPC endpoint (resource endpoint). A resource endpoint lets you privately and securely access VPC resources such as a database, an Amazon EC2 instance, an application endpoint, a domain-name target, or an IP address that may be in a private subnet in another VPC or in an on premise environment. Without resource endpoints, you have to either add an internet gateway to your VPC or access the resource using a AWS PrivateLink interface endpoint and a Network Load Balancer. Resource endpoints don't require a load balancer, so you can access the VPC resource directly. A VPC resource is represented by a resource configuration. A resource configuration is associated with a resource gateway.

Pricing

When you access resources using resource endpoints, you are billed for each hour that your resource VPC endpoint is provisioned. You are also billed per GB of data processed when you access resources. For more information, see AWS PrivateLink pricing. When you enable access to your resources using resource configurations and resource gateways, you are billed per GB data processed by your resource gateways. For more information, see Amazon VPC Lattice pricing.

Contents

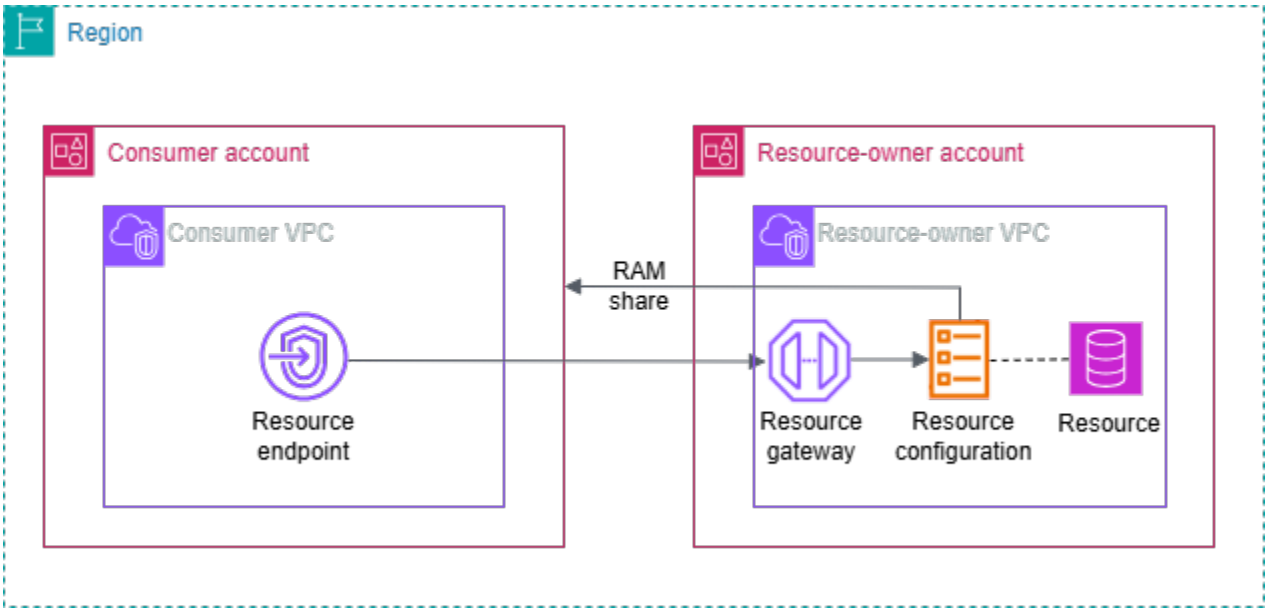
- Overview
- DNS hostnames
- DNS resolution
- Private DNS
- Subnets and Availability Zones
- IP address types
- Access a resource through a resource VPC endpoint
- Manage resource endpoints

Overview

You can access resources in your account or those that have been shared with you from another account. To access a resource, you create a resource VPC endpoint, which establishes connections between the subnets in your VPC and the resource using network interfaces. Traffic destined for

the resource is resolved to the private IP addresses of the resource endpoint's network interfaces using DNS. Then, traffic is sent to the resource using the connection between the VPC endpoint and the resource through the resource gateway.

The following image shows a resource endpoint in a consumer account accessing a resource that is owned by a different account and shared through AWS RAM:



Considerations

- TCP traffic is supported. UDP traffic is not supported.
- Network connections must be initiated from the VPC that contains the resource endpoint, and not from the VPC that has the resource. The resource's VPC can't initiate network connections into the endpoint VPC.
- The only supported ARN-based resources are Amazon RDS resources.
- At least one Availability Zone of the VPC endpoint and the resource gateway have to overlap.

DNS hostnames

With AWS PrivateLink, you send traffic to resources using private endpoints. When you create a resource VPC endpoint, we create Regional DNS names (called default DNS name) that you can use to communicate with the resource from your VPC and from on premises. We recommend that you use DNS instead of endpoint IPs to connect to your resources. The default DNS name for your resource VPC endpoint has the following syntax:

```
endpoint_id.rcfgId.randomHash.vpc-lattice-rsc.region.on.aws
```

When you create a resource VPC endpoint for select resource configurations that use ARNs, you can enable private DNS. With private DNS, you can continue to make requests to the resource using the DNS name provisioned for the resource by the AWS service, while leveraging private connectivity through the resource VPC endpoint. For more information, see the section called "DNS resolution".

The following describe-vpc-endpoint-associations command displays the DNS entries for a resource endpoint.

```
aws ec2 describe-vpc-endpoint-associations --vpc-endpoint-id vpce-123456789abcdefgh --
query 'VpcEndpointAssociations[*].*'
```

The following is example output for a resource endpoint for an Amazon RDS database with private DNS names enabled. The first DNS name is the default DNS name. The second DNS name is from the hidden private hosted zone, which resolves requests to the public endpoint to the private IP addresses of the endpoint network interfaces.

```
[
    [
        "vpce-rsc-asc-abcd1234abcd",
        "vpce-123456789abcdefgh",
        "Accessible",
        {
            "DnsName": "vpce-1234567890abcdefg-
snra-1234567890abcdefg.rcfg-abcdefgh123456789.4232ccc.vpc-lattice-rsc.us-
east-1.on.aws",
            "HostedZoneId": "ABCDEFGH123456789000"
        },
        {
            "DnsName": "database-5-test.cluster-ro-example.us-
east-1.rds.amazonaws.com",
            "HostedZoneId": "A1B2CD3E4F5G6H8I91234"
        },
        "arn:aws:vpc-lattice:us-east-1:111122223333:resourceconfiguration/
rcfg-1234567890abcdefg",
        "arn:aws:vpc-lattice:us-east-1:111122223333:resourceconfiguration/
rcfg-1234567890xyz"
    ]
]
```

DNS resolution

The DNS records that we create for your resource 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 resource endpoint's network interfaces. You can use these DNS names to access the resource from on premises, as long as you have access to the VPC that the resource endpoint is in, through VPN or Direct Connect.

Private DNS

If you enable private DNS for your resource VPC endpoint for select resource configurations that use ARNs, and your VPC has both DNS hostnames and DNS resolution enabled, we create hidden, AWS-managed private hosted zones for resource configurations with a custom DNS name. The hosted zone contains a record set for the default DNS name for the resource that resolves it to the private IP addresses of the resource endpoint's network interfaces in your VPC.

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 the custom DNS name or 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. In a production environment, for high availability and resiliency, we recommend configuring at least two Availability Zones for each VPC endpoint.

IP address types

Resource endpoints can support IPv4, IPv6, or dualstack addresses. Endpoints that support IPv6 can respond to DNS queries with AAAA records. The IP address type of a resource endpoint must be compatible with the subnets for the resource 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 a resource VPC endpoint supports IPv4, the endpoint network interfaces have IPv4 addresses. If a resource 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.

Access a resource through a resource VPC endpoint

You can access a VPC resource such as a domain name, an IP address, or Amazon RDS database using a resource endpoint. A resource endpoint provides private access to a resource. When you create the resource endpoint, you specify a resource configuration of type single, group, or ARN. A resource endpoint can be associated with only one resource configuration. The resource configuration can represent a single resource or a group of resources.

Prerequisites

To create a resource endpoint, you must meet the following prerequisites.

- You must have a resource configuration that you created or another account created and shared with you through AWS RAM.
- If a resource configuration is shared with you from another account, you must review and accept the resource share that contains the resource configuration. For more information, see Accepting and rejecting invitations in the *AWS RAM User Guide*.

VPC Lattice does not create a resource endpoint if the following are true:

- Resource gateway is in the same VPC as the resource endpoint.
- For domain-name targets
  - DNS resolution is set to IN_VPC on the resource gateway.

- The custom domain name or group domain is the same or higher-level domain of the domain-name target.

Create a VPC resource endpoint

Use the following procedure to create a VPC resource endpoint. After you create a resource endpoint, you can only modify its security groups or tags.

To create a VPC resource endpoint

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. You can specify a name to make it easier to find and manage the endpoint.
5. For **Type**, choose **Resources**.
6. For **Resource configurations**, select the resource configuration.
7. For **Network settings**, select the VPC from which you'll access the resource.
8. If, you want to configure private DNS support for resource configurations, select **Additional settings**, **Enable DNS name**. To use this feature, ensure that the attributes **Enable DNS hostnames** and **Enable DNS support** are enabled for your VPC. For more information, see the section called “Custom domain names for resource consumers”.
9. For **Subnets**, select a subnet to create the endpoint network interface in.

   In a production environment, for high availability and resiliency, we recommend configuring at least two Availability Zones for each VPC endpoint.
10. For **Security groups**, select a security group.

    If you do not specify a security group, we associate the default security group for the VPC.
11. Choose **Create endpoint**.

To create a resource endpoint using the command line

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

Manage resource endpoints

After you create a resource endpoint, you can manage its security groups or tags.

Tasks

- Delete an endpoint
- Update an endpoint

Delete an endpoint

When you are finished with a VPC endpoint, you can delete it.

To delete an 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 endpoint.
4. Choose **Actions**, **Delete VPC endpoints**.
5. When prompted for confirmation, enter **`delete`**.
6. Choose **Delete**.

To delete an endpoint using the command line

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

Update an endpoint

You can update a VPC endpoint.

To update an 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 endpoint.

4. Choose **Actions**, and the appropriate option.
5. Follow the console steps to submit the update.

To update an endpoint using the command line

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

Access network segments through AWS PrivateLink

You can privately access a network segment in another VPC using a tunnel VPC endpoint (tunnel endpoint). A tunnel endpoint lets you privately and securely access resources located within a set of CIDR ranges in another network. A CIDR range represents a network segment, so a consumer application can reach any resource within that network segment, without the provider having to enumerate each resource. This is useful when the resources are ephemeral or not known in advance. Tunnel endpoints require you to tunnel into another VPC using GENEVE encapsulation.

Pricing

When you access a network segment using a tunnel endpoint, you are billed for each hour that your tunnel endpoint is provisioned. You are also billed per GB of data processed when you access the network segment. For more information, see AWS PrivateLink pricing. When you share your network segment using resource configurations and resource gateways, you are billed per GB of data processed by your resource gateways. For more information, see Amazon VPC Lattice pricing.

Topics

- Overview
- GENEVE encapsulation
- DNS hostnames
- DNS resolution
- Private DNS
- Subnets and Availability Zones
- IP address types
- Access a network segment through a tunnel VPC endpoint

Overview

You can access network segments in your account or those that have been shared with you from another account. To access a network segment, you first create a tunnel endpoint, which establishes a connection between your VPC and a resource gateway in the VPC from which the network segment is shared. You then encapsulate application traffic using GENEVE and send it to the tunnel endpoint. The endpoint removes the GENEVE header and sends the inner packet to its destination in the other VPC. When doing so, it checks whether the destination lies within

the network segment that was shared with you from the other VPC. If it lies outside, the endpoint doesn't allow access to it.

Considerations

- The subnets of the tunnel endpoint must be in Availability Zones that overlap with the Availability Zones of the subnets in which the resource gateway is.
- Network connections can only be initiated from the VPC that has the tunnel endpoint, and not the VPC that is sharing the network segments. The network segment VPC can't initiate network connections into the endpoint VPC.

GENEVE encapsulation

Application traffic must be encapsulated in GENEVE. GENEVE traffic must use UDP and port 6081 on the tunnel endpoint. The VNI must be set to `0`. The inner packet must be a layer 3 IP packet. In the inner packet, UDP is supported only for DNS queries. Regular application traffic in the inner packet must be TCP.

DNS hostnames

A tunnel endpoint has a regional DNS name. It resolves to IPs in the subnets where the tunnel endpoint is created. These IPs should be the target of your GENEVE-encapsulated traffic, not the application traffic. Private DNS names are not supported for tunnel endpoints.

DNS resolution

A tunnel endpoint lets you resolve DNS in the context of the VPC that is sharing the network segment. You can resolve a domain through the tunnel endpoint against the DNS resolver of the remote VPC. To do so, you use `169.254.168.253` as the name server. You set it as the destination of the DNS query in the inner packet, and encapsulate it in GENEVE like you do for application traffic. DNS resolution looks the same as it would for a client in the remote VPC. It is based on the remote VPC's configured DHCP option set. It respects Route 53 private hosted zones or custom resolvers configured in the remote VPC.

DNS queries sent to `169.254.168.253` are subject to the `IN_VPC` DNS resolution constraints of the resource gateway. For more information about DNS resolution flags, see the section called “Resource Config DNS Resolution”.

Private DNS

Private DNS names are not supported for tunnel endpoints.

Subnets and Availability Zones

You can configure your tunnel endpoint with one subnet per Availability Zone. Elastic network interfaces are created for the endpoint in those subnets. An IP address is assigned to each ENI from its subnet, based on the IP address type of the endpoint. In a production environment, for high availability and resiliency, we recommend configuring at least two Availability Zones for each endpoint. The subnets of the endpoint must be in Availability Zones that overlap with the Availability Zones of the subnets in which the resource gateway is.

IP address types

The outer and inner packets can have differing IP types. The following combinations are supported: IPv4 over IPv4, IPv6 over IPv4, IPv4 over IPv6, and IPv6 over IPv6.

The tunnel endpoint's IP type decides the IP type of the outer GENEVE packet. The remote VPC's IP type decides the inner packet's IP type. A network segment that is shared can have both IPv4 and IPv6 ranges.

Tunnel endpoints support IPv4, IPv6, or dualstack addresses. The IP address type of a tunnel endpoint must be compatible with the subnets for the tunnel endpoint, as described here:

- IPv4 – Assigns IPv4 addresses to the tunnel endpoint. Supported only if all of the endpoint's subnets have IPv4 ranges.
- IPv6 – Assigns IPv6 addresses to the tunnel endpoint. Supported only if all of the endpoint's subnets have IPv6 ranges.
- Dualstack – Assigns both IPv4 and IPv6 addresses to the tunnel endpoint. Supported only if all of the endpoint's subnets have both IPv4 and IPv6 ranges.

The endpoint's IPv6 addresses are unreachable from the internet. `denyAllIgwTraffic` is enabled on its network interfaces.

Access a network segment through a tunnel VPC endpoint

You can access a network segment through a tunnel endpoint. A network segment is a set of CIDR ranges. It is shared as a resource configuration of type CIDR. A tunnel endpoint associates with exactly one CIDR-type resource configuration.

Prerequisites

To create a tunnel endpoint, you must meet the following prerequisites.

- You must have a CIDR-type resource configuration that you created, or that another account created and shared with you through AWS RAM.
- If a resource configuration is shared with you from another account, you must review and accept the resource share that contains the resource configuration. For more information, see Accepting and rejecting invitations in the *AWS RAM User Guide*.
- The resource configuration must be attached to a resource gateway whose DNS resolution flag is set to `IN_VPC`.
- Your consumer VPC must have subnets in Availability Zones that overlap the provider resource gateway's Availability Zones.
- Your consumer compute must be able to encapsulate application traffic in GENEVE. The VNI must be set to `0`. The inner packet must be a layer 3 IP packet. This should ideally be done in its own routing context or network namespace.
- Security groups on both the endpoint and the compute must allow UDP port 6081.

Create a tunnel VPC endpoint

After you create a tunnel endpoint, you must describe its association to obtain the per-Availability Zone endpoint IP addresses, and then configure the GENEVE device and the tunnel nameserver on your compute.

To create a tunnel endpoint (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. (Optional) Enter a name to make it easier to find and manage the endpoint.

5. For **Type**, choose the tunnel endpoint type.
6. For **Resource configurations**, select the CIDR resource configuration.
7. For **Network settings**, select the VPC from which you'll access the range.
8. For **Subnets**, select a subnet in each Availability Zone that overlaps the resource gateway's Availability Zones.
9. For **Security groups**, select a security group that allows UDP port 6081. If you do not specify a security group, we associate the default security group for the VPC.
10. Choose **Create endpoint**.

The following create-vpc-endpoint command creates a tunnel endpoint for a CIDR resource configuration:

```
aws ec2 create-vpc-endpoint \
    --vpc-endpoint-type Tunnel \
    --vpc-id vpc-1a2b3c4d \
    --subnet-ids subnet-1a2b3c4d \
    --ip-address-type ipv4 \
    --resource-configuration-arn arn:aws:vpc-
lattice:region:111122223333:resourceconfiguration/rcfg-1234567890abcdefg
```

To create a tunnel endpoint using the command line

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

Access service networks through AWS PrivateLink

You can privately connect to a service network from your VPC using a service network VPC endpoint (service-network endpoint). A service-network endpoint lets you privately and securely access the resources and services that are associated to the service network. In this way, you can privately access multiple resources and services through a single VPC endpoint.

A service network is a logical collection of resource configurations and VPC Lattice services. Using a service-network endpoint, you can connect a service network to your VPC, and access those resources and services privately from your VPC or from on-premises. A service-network endpoint lets you connect to one service network. To connect to multiple service networks from your VPC, you can create multiple service-network endpoints, each pointing to a different service network.

Service networks are integrated with AWS Resource Access Manager (AWS RAM). You can share your service network with another account through AWS RAM. When you share a service network with another AWS account, that account can create a service-network endpoint to connect to the service network. You can share a service network using a resource share in AWS RAM.

Use the AWS RAM console, to view the resource shares to which you have been added, the shared service networks that you can access, and the AWS accounts that have shared the resources with you. For more information, see Resources shared with you in the *AWS RAM User Guide*.

Pricing

You are billed hourly for the resource configurations that are associated with your service network. You are also billed per GB of data processed when you access resources through the service network VPC endpoint. You are not billed hourly for the service-network VPC endpoint itself. For more information, see Amazon VPC Lattice pricing.

Contents

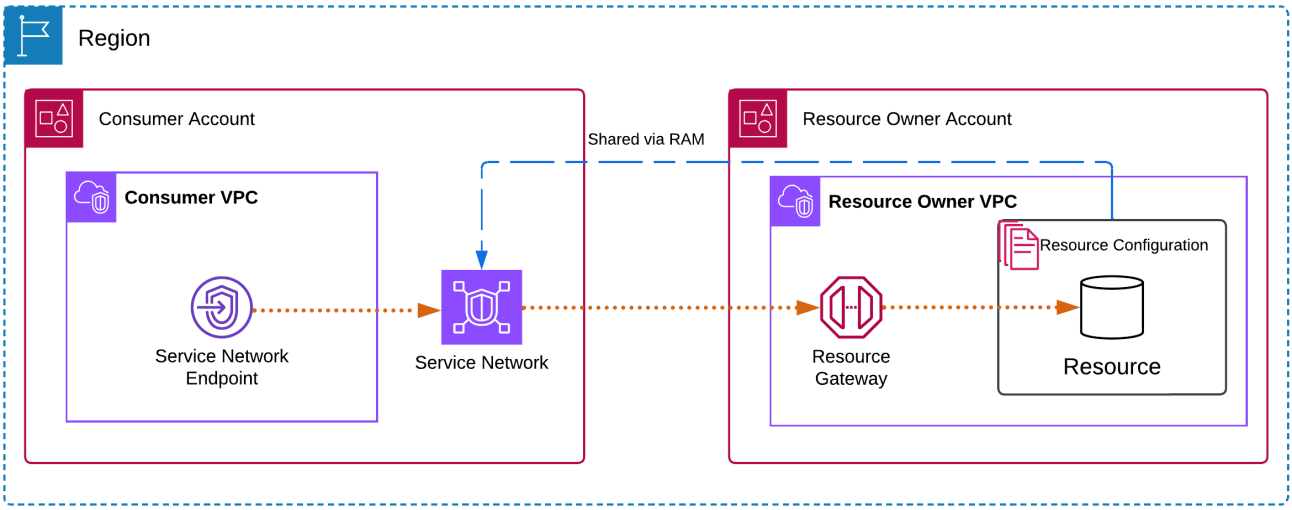
- Overview
- DNS hostnames
- DNS resolution
- Private DNS
- Subnets and Availability Zones
- IP address types
- Access a service network through a service-network endpoint

- Manage service-network endpoints

Overview

You can either create your own service network, or a service network can be shared with you from another account. Either way, you can create a service-network endpoint to connect to it from your VPC. For more information on how to create service network and associate resource configurations to it, see the Amazon VPC Lattice User Guide.

The following diagram shows how a service-network endpoint in your VPC accesses a service network.



Network connections can only be initiated from the VPC that has the service-network endpoint to the resources and services in the service network. The VPC with the resources and services can't initiate network connections into the endpoint VPC.

DNS hostnames

With AWS PrivateLink, you send traffic to service networks using private endpoints. When you create a service-network VPC endpoint, we create Regional DNS names (called default DNS name) for each resource and service that you can use to communicate with the resource and service from your VPC and from on premises. IP addresses associated with the endpoint can change. We recommend that you use DNS instead of endpoint IPs to connect to your service networks.

The default DNS name for a resource in the service network has the following syntax:

```
endpointId-snraId.rcfgId.randomHash.vpc-lattice-rsc.region.on.aws
```

The default DNS name for a Lattice service in the service-network has the following syntax:

```
endpointId-snsaId.randomHash.vpc-lattice-svcs.region.on.aws
```

If you're using the AWS Management Console, you can find the DNS name under the **Associations** tab. If you're using the AWS CLI, use the describe-vpc-endpoint-associations command.

You can only enable private DNS when your service network has an ARN-type resource configuration to an Amazon RDS database service. With private DNS, you can continue to make requests to the resource using the DNS name provisioned for the resource by the AWS service, while leveraging private connectivity through the service-network VPC endpoint. For more information, see the section called “DNS resolution”.

DNS resolution

When you create a service network endpoint, we create DNS names for each resource configuration and Lattice service that is associated to the service network. These DNS records are public. Therefore, these DNS names are publicly resolvable. However, DNS requests from outside the VPC still return the private IP addresses of the service network endpoint’s network interfaces. You can use these DNS names to access the resource and services from on premises, as long as you have access to the VPC that the service network endpoint is in, through VPN or Direct Connect.

Private DNS

If you enable private DNS for your service-network VPC endpoint, and your VPC has both DNS hostnames and DNS resolution enabled, we create hidden, AWS-managed private hosted zones for the resource configurations that have custom DNS names. The hosted zone contains a record set for the default DNS name for the resource that resolves it to the private IP addresses of the service-network endpoint's network interfaces in your VPC.

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 the default DNS names or 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 service-network endpoint with one subnet per Availability Zone. We create an elastic network interface for the VPC endpoint in each subnet that you specify. We assign IP addresses to each elastic network interface from its subnet as follows:

- **VPC Lattice services (Layer 7)** – We assign a /28 block (16 contiguous IPv4 addresses) per Availability Zone for all VPC Lattice services associated with the service network. This /28 block is allocated when the service-network endpoint is created, even if there are no services currently in the service network. The /28 block must consist of 16 contiguous, unoccupied IPv4 addresses and it cannot overlap with the five AWS-reserved addresses (first four and last IP). Ensure that sufficient free contiguous address space is available. For IPv6, we also assign a /80 block per Availability Zone for VPC Lattice services.
- **VPC Lattice resources (Layer 4/TCP)** – We assign one IPv4 address per resource configuration per Availability Zone. Contiguous address space is not required for VPC Lattice resources. We allocate up to 63 IP addresses per elastic network interface. When additional resource configurations exceed this limit, we create another elastic network interface in the same subnet. For IPv6, we assign a /80 block on the first elastic network interface created for resources; no additional elastic network interfaces are created when using IPv6. When you remove a resource configuration from the service network, we release its associated IP address. When all IPv4 addresses on an elastic network interface are released, we remove the elastic network interface.

In a production environment, for high availability and resiliency, we recommend that you configure at least two Availability Zones for each service-network endpoint and ensure that each subnet has sufficient available IPv4 addresses.

IP address types

Service-network endpoints can support IPv4, IPv6, or dual-stack addresses. Endpoints that support IPv6 can respond to DNS queries with AAAA records. The IP address type of a service-network endpoint must be compatible with the subnets for the resource 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 a service-network VPC endpoint supports IPv4, the endpoint network interfaces have IPv4 addresses. If a service-network 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.

Access a service network through a service-network endpoint

You can access a service network using a service-network endpoint. A service-network endpoint provides private access to resource configurations and services in the service network.

Prerequisites

To create a service-network endpoint, you must meet the following prerequisites.

- You must have a service network that was either created by you or shared with you from another account through AWS RAM.
- If a service network is shared with you from another account, you must review and accept the resource share that contains the service network. For more information, see Accepting and rejecting invitations in the *AWS RAM User Guide*.
- For VPC Lattice services associated with the service network, the service-network endpoint requires a contiguous /28 block (16 IPv4 addresses) per Availability Zone. This /28 block is allocated when the endpoint is created, even if no services are currently in the service network. The /28 block must consist of 16 contiguous, unoccupied IPv4 addresses and cannot overlap with the five AWS-reserved addresses (first four and last IP in the subnet). For IPv6, a /80 block per Availability Zone is allocated for VPC Lattice services. Ensure that sufficient free contiguous address space is available in each subnet you select.
- For VPC Lattice resources (Layer 4/TCP) associated with the service network, one IPv4 address per resource configuration per Availability Zone is required. Contiguous address space is not required. Up to 63 IP addresses can be allocated per elastic network interface. When additional resource configurations exceed this limit, an additional elastic network interface is created in the same subnet. For IPv6, a /80 block is assigned on the first elastic network interface created for resources; no additional elastic network interfaces are created when using IPv6.

If you need to avoid consuming VPC CIDR IP addresses, or anticipate a large number of resource configurations associated with the service network, consider using a service network VPC association instead. For more information, see Manage VPC endpoint associations in the *Amazon VPC Lattice User Guide*.

Create a service network endpoint

Create a service-network endpoint to access the service network that was shared with you. After you create a service-network endpoint, you can only modify its security groups or tags.

To create a service-network endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, under **PrivateLink and Lattice**, choose **Endpoints**.
3. Choose **Create endpoint**.
4. You can specify a name to make it easier to find and manage the endpoint.
5. For **Type**, choose **Service networks**.
6. For **Service networks**, select the service network.
7. For **Network settings**, select your VPC from which you'll access the service network.
8. If, you want to configure private DNS support, select **Additional settings**, **Enable private DNS name**. To use this feature, ensure that the attributes **Enable DNS hostnames** and **Enable DNS support** are enabled for your VPC.
9. For **Subnets**, select a subnet to create the endpoint network interface in.

   In a production environment, for high availability and resiliency, we recommend configuring at least two Availability Zones for each VPC endpoint.
10. For **Security groups**, select a security group.

    If you do not specify a security group, we associate the default security group for the VPC.
11. Choose **Create endpoint**.

To create a service-network endpoint using the command line

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

Manage service-network endpoints

After you create a service-network endpoint, you can update its security groups or tags.

Tasks

- Delete an endpoint
- Update a service-network endpoint

Delete an endpoint

When you are finished with a VPC endpoint, you can delete it.

To delete an 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 service-network endpoint.
4. Choose **Actions**, **Delete VPC endpoints**.
5. When prompted for confirmation, enter **`delete`**.
6. Choose **Delete**.

To delete an endpoint using the command line

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

Update a service-network endpoint

You can update a VPC endpoint.

To update an 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 endpoint.

4. Choose **Actions**, and the appropriate option.
5. Follow the console steps to submit the update.

To update an endpoint using the command line

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

Resource configuration for VPC resources

A resource configuration represents an individual resource, a group of resources, or a list of CIDR ranges that you want to make accessible to clients in other VPCs and accounts. By defining a resource configuration, you can allow private, secure, unidirectional network connectivity to resources in your VPC from clients in other VPCs and accounts. A resource configuration is associated with a resource gateway through which it receives traffic.

Contents

- Types of resource configurations
- Resource gateway
- Custom domain names for resource providers
- Custom domain names for resource consumers
- Custom domain names for service network owners
- Resource definition
- Protocol
- Port ranges
- Accessing resources
- Association with service network type
- Types of service networks
- Sharing resource configurations through AWS RAM
- Monitoring
- Create a resource configuration in VPC Lattice
- Manage associations for a VPC Lattice resource configuration

Types of resource configurations

A resource configuration can be of several types. The different types help represent different kinds of resources. The types are:

- **Single resource configuration**: An IP address or a domain name. It can be shared independently.
- **Group resource configuration**: A collection of child resource configurations. It can be shared independently.

- **Child resource configuration**: A member of a Group resource configuration. It represents an IP address or a domain name. It can't be shared independently, and can only be shared as part of a group. It can be added and removed from a group seamlessly. When added, it's automatically accessible to those who can access the group.
- **ARN resource configuration**: A supported resource type that is provisioned by an AWS service. For example, an Amazon RDS database. Child resource configurations are automatically managed by AWS. It can be shared independently.
- **CIDR resource configuration**: A list of CIDR ranges. It can be shared independently.

Resource gateway

A resource configuration is associated with a resource gateway. A resource gateway is a set of ENIs that serve as a point of ingress into the VPC in which the resource is in. Multiple resource configurations can be associated with the same resource gateway. When clients in other VPCs or accounts access a resource in your VPC, the resource sees traffic coming locally from the resource gateway in that VPC.

Custom domain names for resource providers

Resource providers can attach a custom domain name to a resource configuration, such as `example.com`, which resource consumers can use to access the resource configuration. The custom domain name can be owned and verified by the resource provider, or it can be a third-party or AWS domain. Resource providers can use resource configurations to share cache clusters and Kafka clusters, TLS-based applications, or other AWS resources.

The following considerations apply to providers of resource configurations:

- A resource configuration can only have one custom domain.
- The custom domain name of a resource configuration cannot be changed.
- The custom domain name is visible to all resource configuration consumers.
- You can verify your custom domain name using the domain name verification process in VPC Lattice. For more information, see https://docs.aws.amazon.com/vpc-lattice/latest/ug/create-and-verify.html.
- For resource configurations of type group and child, you must first specify a group domain on the group resource configuration. Afterward, the child resource configurations can have custom domains that are subdomains of the group domain. If the group doesn't have a group domain,

you can use any custom domain name for the child, but VPC Lattice will not provision any hosted zones for the child domain names in the resource consumer's VPC.

- Custom domain names are not supported for CIDR resource configurations.

Custom domain names for resource consumers

When resource consumers enable connectivity to a resource configuration that has a custom domain name, they can allow VPC Lattice to manage a Route 53 private hosted zone in their VPC. Resource consumers have granular options for which domains they want to allow VPC Lattice to manage private hosted zones for.

Resource consumers can set the `private-dns-enabled` parameter when enabling connectivity to resource configurations through a resource endpoint, a service network endpoint, or a service network VPC association. Along with the `private-dns-enabled` parameter, consumers can use DNS options to specify which domains that they want VPC Lattice to manage private hosted zones for. Consumers can choose between the following private DNS preferences:

`ALL_DOMAINS`

VPC Lattice provisions private hosted zones for all custom domain names.

`VERIFIED_DOMAINS_ONLY`

VPC Lattice provisions a private hosted zone only if the custom domain name has been verified by the provider.

`VERIFIED_DOMAINS_AND_SPECIFIED_DOMAINS`

VPC Lattice provisions private hosted zones for all verified custom domain names and other domain names that the resource consumer specifies. The resource consumer specifies the domain names in the `private DNS specified domains` parameter.

`SPECIFIED_DOMAINS_ONLY`

VPC Lattice provisions a private hosted zone for domain names specified by the resource consumer. The resource consumer specifies the domain names in the `private DNS specified domains` parameter.

When you enable private DNS, VPC Lattice creates a private hosted zone in your VPC for the custom domain name associated with the resource configuration. By default, the private DNS preference is set to `VERIFIED_DOMAINS_ONLY`. This means that private hosted zones are

created only if the custom domain name has been verified by the resource provider. If you set your private DNS preference to `ALL_DOMAINS` or `SPECIFIED_DOMAINS_ONLY` then VPC Lattice creates private hosted zones regardless of the verification status of the custom domain name. When a private hosted zone is created for a given domain, all traffic to that domain from your VPC is routed through VPC Lattice. We recommend that you use the `ALL_DOMAINS`, `VERIFIED_DOMAINS_AND_SPECIFIED_DOMAINS`, or `SPECIFIED_DOMAINS_ONLY` preferences only when you want traffic to these custom domain names to go through VPC Lattice.

We recommend that resource consumers set their private DNS preference to `VERIFIED_DOMAINS_ONLY`. This lets consumers tighten their security perimeter by only allowing VPC Lattice to provision private hosted zones for verified domains in the resource consumer's account.

To select domains in the private DNS specified domains, resource consumers can enter a fully qualified domain name, such as `my.example.com` or use a wildcard such as `*.example.com`.

The following considerations apply to consumers of resource configurations:

- The private DNS enabled parameter cannot be changed.
- Private DNS should be enabled on a service network resource association for private hosted zones to be created in a VPC. For a resource configuration, the private DNS enabled status of the service network resource association overrides the private DNS enabled status of either the service network endpoint or service network VPC association.

For resource configurations that are domain-name targets, a private hosted zone entry is not created if the following are true:

- Resource gateway is in the same VPC as the service network VPC endpoint/service network VPC association.
- DNS resolution is set to IN_VPC on the resource gateway.
- The custom domain name or group domain is the same or higher-level domain of the domain-name target.

For resource configurations that are of type ARN, VPC Lattice does not create a private hosted zone entry if the following are true:

- Resource gateway is in the same VPC as the service network VPC endpoint/service network VPC association.

Custom domain names for service network owners

The private DNS enabled property of the service network resource association overrides the private DNS enabled property of the service network endpoint and the service network VPC association.

If a service network owner creates a service network resource association and doesn't enable private DNS, VPC Lattice won't provision private hosted zones for that resource configuration in any VPCs that the service network is connected to, even though private DNS is enabled on the service network endpoint or service network VPC associations.

For resource configurations of type ARN, the private DNS flag is true and immutable. Therefore, VPC Lattice provisions private hosted zones for ARN resource types regardless of the private DNS property setting of the service network endpoint and the service network VPC association, except when the resource gateway is also in the same VPC. In other words, when a VPC is both a consumer and a provider for an ARN type of resource configuration, VPC Lattice skips the creation of the private hosted zones in such a VPC.

Resource definition

In the resource configuration, identify the resource in one of the following ways:

- By an **Amazon Resource Name (ARN)**: Supported resource types that are provisioned by AWS services can be identified by their ARN. Only Amazon RDS databases are supported. You can't create a resource configuration for a publicly accessible cluster.
- By a **domain-name target**: You can use any domain name. If you use a private DNS server or your domain is in a Route 53 private hosted zone, then the resource gateway must have DNS resolution set to IN_VPC. If your domain name points to an IP that's outside of your VPC, you must have a NAT gateway in your VPC. Domain-name targets that resolve to public IPv6 addresses are not supported.
- By an **IP address**: For IPv4, specify a private IP from the following ranges: 10.0.0.0/8, 100.64.0.0/10, 172.16.0.0/12, 192.168.0.0/16. For IPv6, specify an IP from the VPC. Public IPs aren't supported.
- By a **CIDR range**: For a resource configuration of type CIDR, specify the list of CIDRs you want to share (for example, 10.0.0.0/24). The CIDRs should be reachable from the resource gateway. To provide full network access, specify 0.0.0.0/0 if using IPv4 or ::/0 if using IPv6.

Protocol

When you create a resource configuration you can define the protocols that the resource will support. Currently, only the TCP protocol is supported.

For CIDR resource configurations, the protocol has to be TCP_UDP: only TCP is supported for application traffic, UDP is supported only for DNS queries.

Port ranges

When you create a resource configuration you can define the ports it will accept requests on. Client access on other ports will not be allowed.

Accessing resources

Consumers can access resource configurations directly from their VPC using a VPC endpoint or through a service network. As a consumer, you can enable access from your VPC to a resource configuration that is in your account or that has been shared with you from another account through AWS RAM.

- *Accessing a resource configuration directly*

  You can access resource configurations directly through AWS PrivateLink-based resource endpoints and tunnel endpoints. You can create a resource endpoint in your VPC to access resource configurations of types ARN, Single, and Group, and can create a tunnel endpoint to access resource configurations of type CIDR. For more information on how to create resource and tunnel endpoints, see Accessing VPC resources and Accessing network segments in the *AWS PrivateLink user guide*.

- *Accessing a resource configuration through a service network*

  You can associate a resource configuration with a service network, and connect your VPC to the service network. You can connect your VPC to the service network either through an association or using a AWS PrivateLink-based service network endpoint. CIDR resource configurations cannot be associated with a service network.

  For more information on service network associations, see Manage the associations for a VPC Lattice service network.

For more information on service network VPC endpoints, see Access service networks in the *AWS PrivateLink user guide*.

When private DNS is enabled for your VPC, you can't create a resource endpoint and service network endpoint for the same resource configuration.

Association with service network type

When you share a resource configuration with a consumer account, for example, Account-B, through AWS RAM, Account-B can access the resource configuration either directly through a resource endpoint or tunnel endpoint, or through a service network. CIDR resource configurations cannot be accessed through a service network. They can only be accessed directly through a tunnel endpoint.

To access a resource configuration through a service network, Account-B would have to associate the resource configuration with a service network. Service networks are shareable between accounts. So, Account-B can share their service network (that the resource configuration is associated to) with Account-C, making your resource accessible from Account-C.

To prevent such transitive sharing, you can specify that your resource configuration cannot be added to service networks that are shareable between accounts. If you specify this, then Account-B won't be able to add your resource configuration to service networks that are shared or can be shared with another account in the future.

Types of service networks

When you share a resource configuration with another account, for example Account-B, through AWS RAM, Account-B can access the resource in one of four ways:

- Using a AWS PrivateLink-based resource endpoint (for Single, Group, and ARN resource configurations).
- Using a AWS PrivateLink-based tunnel endpoint (for CIDR resource configurations).
- Using a AWS PrivateLink-based service network endpoint.
- Using a service network VPC association.

When you use a service-network association, each resource is assigned an IP per subnet from the 129.224.0.0/17 block, which is AWS owned and non-routable. This is in addition to the managed prefix list that VPC Lattice uses to route traffic to services over the VPC Lattice network. Both of these IPs are updated to your VPC route table.

For service network endpoint and service network VPC association, the resource configuration would have to be put in a service network in Account-B. Service networks are shareable between accounts. So, Account-B can share their service network (that contains the resource configuration) with Account-C, making your resource accessible from Account-C. To prevent such transitive sharing, you can disallow your resource configuration from being added to service networks that are shareable between accounts. If you disallow this, then Account-B won't be able to add your resource configuration to a service network that is shared or can be shared with another account.

Sharing resource configurations through AWS RAM

Resource configurations are integrated with AWS Resource Access Manager. You can share your resource configuration with another account through AWS RAM. When you share a resource configuration with an AWS account, clients in that account can privately access the resource. You can share a resource configuration using a resource share in AWS RAM.

Use the AWS RAM console, to view the resource shares to which you have been added, the shared resources that you can access, and the AWS accounts that have shared resources with you. For more information, see Resources shared with you in the *AWS RAM User Guide*.

To access a resource from another VPC in the same account as the resource configuration, you don't need to share the resource configuration through AWS RAM.

Monitoring

You can enable monitoring logs on your resource configuration. You can choose a destination to send the logs to.

Create a resource configuration in VPC Lattice

Create a resource configuration.

AWS Management Console

To create a resource configuration using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, under **PrivateLink and Lattice**, choose **Resource configurations**.
3. Choose **Create resource configuration**.
4. Enter a name that is unique within your AWS account. You can't change this name after the resource configuration is created.
5. For **Configuration type**, choose **Resource** for a single or child resource or **Resource group** for a group of child resources.
6. Choose a resource gateway that you previously created or create one now.
7. (Optional) To enter a custom domain name, do one of the following:
   - If you have a resource configuration of type single, you can enter a custom domain name. Resource consumers can use this domain name to access your resource configurations.
   - If you have a resource configuration of type group and child, you must first specify a group domain on the group resource configuration. Next, the child resource configurations can have custom domains that are subdomains of the group domain.
8. (Optional) Enter the verification ID.

   Provide a verification ID if you want your domain name to be verified. This lets resource consumers know that you own the domain name.
9. Choose the identifier for the resource that you want this resource configuration to represent.
10. Choose the port ranges through which you want to share the resource.
11. For **Association settings**, specify whether this resource configuration can be associated with shareable service networks.
12. For **Share resource configuration**, choose the resource shares that identify the principals who can access this resource.
13. (Optional) For **Monitoring**, enable **Resource access logs** and the delivery destination if you want to monitor requests and responses to and from the resource configuration.
14. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
15. Choose **Create resource configuration**.

AWS CLI

The following create-resource-configuration command creates a single resource configuration and associates it with the custom domain name `example.com`.

```
aws vpc-lattice create-resource-configuration \
    --name my-resource-config \
    --type SINGLE \
    --resource-gateway-identifier rgw-0bba03f3d56060135 \
    --resource-configuration-definition 'ipResource={ipAddress=10.0.14.85}' \
    --custom-domain-name example.com \
    --verification-id dv-aaaa0000000111111
```

The following create-resource-configuration command creates a group resource configuration and associates it with the custom domain name `example.com`.

```
aws vpc-lattice-custom-dns create-resource-configuration \
  --name my-custom-dns-resource-config-group \
  --type GROUP \
  --resource-gateway-identifier rgw-0bba03f3d56060135 \
  --domain-verification-identifier dv-aaaa0000000111111
```

The following create-resource-configuration command creates a child resource configuration and associates it with the custom domain name `child.example.com`.

```
aws vpc-lattice-custom-dns create-resource-configuration \
  --name my-custom-dns-resource-config-child \
  --type CHILD \
  --resource-configuration-definition 'dnsResource={domainName=my-alb-123456789.us-
west-2.elb.amazonaws.com,ipAddressType=IPV4}' \
  --resource-configuration-group-identifier rcfg-07129f3acded87626 \
  --custom-domain-name child.example.com
```

Manage associations for a VPC Lattice resource configuration

Consumer accounts with which you share a resource configuration, and clients in your account, can access the resource configuration either directly using a resource VPC endpoint or through a service network endpoint. As a result, your resource configuration will have endpoint associations and service network associations.

Manage service network resource associations

Create or delete a service network association.

> **Note**
>
> If you receive an access-denied message while creating the association between the service network and resource configuration, check your AWS RAM policy version and ensure that it is version 2. For more information, see the AWS RAM user guide.

To manage a service-network association using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, under **PrivateLink and Lattice**, choose **Resource configurations**.
3. Select the name of the resource configuration to open its details page.
4. Select **Service network associations** tab.
5. Choose **Create associations**.
6. Select a service network from **VPC Lattice service networks**. To create a service network, choose **Create a VPC Lattice network**.
7. (Optional) To add a tag, expand **Service association tags**, choose **Add new tag**, and enter a tag key and tag value.
8. (Optional) To enable private DNS names for this service network resource association choose **enable private DNS name**. For more information, see the section called “Custom domain names for service network owners”.
9. Choose **Save changes**.
10. To delete an association, select the check box for the association and then choose **Actions**, **Delete**. When prompted for confirmation, enter **`confirm`** and then choose **Delete**.

To create a service network association using the AWS CLI

Use the create-service-network-resource-association command.

To delete a service network association using the AWS CLI

Use the delete-service-network-resource-association command.

Manage resource VPC endpoint associations

Consumer accounts with access to your resource configuration or clients in your account can access the resource configuration using a resource VPC endpoint. If your resource configuration has a custom domain name, you can use enable private DNS to allow VPC Lattice to provision private hosted zones for your resource endpoint or service network endpoint. With this, clients can directly curl the domain name to access the resource configuration. For more information, see the section called “Custom domain names for resource consumers”.

AWS Management Console

1. To create a new endpoint association, go to **PrivateLink and Lattice** in the left navigation pane and choose **Endpoints**.
2. Choose **Create endpoints**.
3. Select the resource configuration you want to connect to your VPC.
4. Select the VPC, subnets, and security groups.
5. (Optional) To turn on private DNS and configure DNS options, select **Enable DNS name**.
6. (Optional) To tag your VPC endpoint, choose **Add new tag**, and enter a tag key and tag value.
7. Choose **Create endpoint**.

AWS CLI

The following create-vpc-endpoint command creates a VPC endpoint that uses private DNS. The private DNS preferences are set to `VERIFIED_AND_SELECTED` and the selected domains are `example.com` and `example.org`. VPC Lattice only provisions private hosted zones for any verified domains or `example.com` or `example.org`.

```
aws ec2 create-vpc-endpoint \
  --vpc-endpoint-type Resource \
  --vpc-id vpc-111122223333aabbc \
  --subnet-ids subnet-0011aabbcc2233445 \
  --resource-configuration-arn arn:aws:vpc-lattice:us-
west-2:111122223333:resourceconfiguration/rcfg-07129f3acded87625 \
  --private-dns-enabled \
  --private-dns-preferences VERIFIED_DOMAINS_AND_SPECIFIED_DOMAINS \
  --private-domains-set example.com, example.org
```

To create a VPC endpoint association using the AWS CLI

Use the create-vpc-endpoint command.

To delete a VPC endpoint association using the AWS CLI

Use the delete-vpc-endpoint command.

Resource gateway in VPC Lattice

A resource gateway is a point of inbound traffic into the VPC where a resource resides. It spans multiple Availability Zones.

A VPC must have a resource gateway if you plan to make resources inside the VPC accessible from other VPCs or accounts. Every resource you share is associated with a resource gateway. When clients in other VPCs or accounts access a resource in your VPC, the resource sees traffic coming locally from the resource gateway in that VPC. The source IP of the traffic is the IP address of the resource gateway. You can assign multiple IP addresses to a resource gateway to allow for more network connections with the resource. Multiple resources in a VPC can be associated with the same resource gateway.

A resource gateway does not provide load balancing capabilities.

Contents

- Considerations
- Security groups
- IP address types
- IPv4 addresses per ENI
- Resource Config DNS Resolution
- Create a resource gateway in VPC Lattice
- Delete a resource gateway in VPC Lattice

Considerations

The following considerations apply to resource gateways:

- For your resource to be accessible from all Availability Zones, you should create your resource gateways to span as many Availability Zones as possible.
- At least one Availability Zone of the VPC endpoint and the resource gateway have to overlap.
- A VPC can have a maximum of 100 resource gateways. For more information, see Quotas for VPC Lattice.
- You can't create a resource gateway in a shared subnet.

Security groups

You can attach security groups to a resource gateway. Security group rules for resource gateways control outbound traffic from the resource gateway to resources.

Recommended outbound rules for traffic flowing from a resource gateway to a database resource

For traffic to flow from a resource gateway to a resource, you must create outbound rules for the resource's accepted listener protocols and port ranges.

Destination	Protocol	Port range	Comment
CIDR range for resource	TCP	3306	Allows traffic from resource gateway to databases.

IP address types

A resource gateway can have IPv4, IPv6, or dual-stack addresses. The IP address type of a resource gateway must be compatible with the subnets of the resource gateway and the IP address type of the resource, as described here:

- **IPv4** – Assign IPv4 addresses to your gateway network interfaces. This option is supported only if all selected subnets have IPv4 address ranges, and the resource also has an IPv4 address.
- **IPv6** – Assign IPv6 addresses to your gateway network interfaces. This option is supported only if all selected subnets are IPv6-only subnets, and the resource also has an IPv6 address.
- **Dualstack** – Assign both IPv4 and IPv6 addresses to your gateway network interfaces. This option is supported only if all selected subnets have both IPv4 and IPv6 address ranges, and the resource either has an IPv4 or an IPv6 address.

The IP address type of the resource gateway is independent of the IP address type of the client or the VPC endpoint through which the resource is accessed.

IPv4 addresses per ENI

If your resource gateway has an IPv4 or a dual-stack IP address type, you can configure the number of IPv4 addresses assigned to each ENI of your resource gateway. When you create a resource gateway, you choose from 1 to 62 IPv4 addresses. After you set the number of IPv4 addresses, the value can't be changed.

The IPv4 addresses are used for network address translation and determine the maximum number of concurrent IPv4 connections to a resource. By default, all resource gateways are assigned 16 IPv4 addresses per ENI. This is a suitable number of IPs to form connections with your backend resources.

If your resource gateway uses the IPv6 address type, the resource gateway automatically receives a /80 CIDR per ENI. This value can't be changed.

Resource Config DNS Resolution

You can specify how a resource gateway performs DNS resolution for resource configurations that are domain-name targets. This property is immutable. You can choose:

- **PUBLIC (default)** - Domain names are resolved using public DNS resolvers.
- **IN_VPC** - Domain names are resolved using the DNS server configured in the Dynamic Host Configuration Protocol (DHCP) option set of the VPC which the resource gateway is in. You should use this if you are using a private DNS server or your domain-name targets are in a Route 53 private hosted zone.

The following table lists the constraints that apply when DNS resolution is set to IN_VPC.

Constraint	Details
IPv6 only subnet support	You can't attach an `IN_VPC` enabled resource gateway to an IPv6 only subnet.
ARN resource configurations	You can't attach resource configurations defined by ARN to the resource gateway.
DNS transport protocol	DNS queries use UDP only. TCP fallback isn't supported.

Constraint	Details
Maximum DNS payload size	VPC Lattice advertises a maximum DNS payload size of 8,400 bytes. Because TCP fallback isn't supported, DNS resolution fails for responses larger than this size.
Recursive DNS resolver	The DNS resolver must be recursive. If a response contains a CNAME record, the response must also contain the records that the CNAME resolves to.

To attach a CIDR resource configuration to a resource gateway, it must have `IN_VPC` enabled.

DHCP option set DNS server changes take 24 hours to propagate to a resource gateway that uses `IN_VPC` DNS resolution.

Create a resource gateway in VPC Lattice

Use the console to create a resource gateway.

To create a resource gateway using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, under **PrivateLink and Lattice**, choose **Resource gateways**.
3. Choose **Create resource gateway**.
4. Enter a name that is unique within your AWS account.
5. Choose the type of IP address for the resource gateway.
6. For **IP address type**, choose the IP address type for the resource gateway.
    - If you selected **IPv4** or **Dualstack** for the **IP address type**, you can enter the number of IPv4 addresses per ENI for your resource gateway.

      The default is 16 IPv4 addresses per ENI. This is a suitable number of IPs to form connections with your backend resources.
7. Choose the VPC that the resource is in.
8. For **Security groups**, choose up to five security groups to control inbound traffic from the VPC to the service network.

9. For **Resource Config DNS Resolution**, choose how you want DNS to be resolved for domain-name targets.
    - If you are using a private DNS server or your domain-name targets are in a Route 53 private hosted zone, set DNS resolution to IN_VPC.
10. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
11. Choose **Create resource gateway**.

To create a resource gateway using the AWS CLI

Use the create-resource-gateway command.

Delete a resource gateway in VPC Lattice

Use the console to delete a resource gateway.

To delete a resource gateway using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, under **PrivateLink and Lattice**, choose **Resource gateways**.
3. Select the check box for the resource gateway that you want to delete and choose **Actions**, **Delete**. When prompted for confirmation, enter **`confirm`** and then choose **Delete**.

To delete a resource gateway using the AWS CLI

Use the delete-resource-gateway command.

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
- AWS managed policies for AWS PrivateLink

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 as the AWS account root user, an IAM user, or by assuming an IAM role.

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

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

AWS account root user

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

Federated identity

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

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

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

IAM users and groups

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

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

IAM roles

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

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

Managing access using policies

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

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

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

Identity-based policies

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

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

Resource-based policies

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

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

Other policy types

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

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

Multiple policy types

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

How AWS 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 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 Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

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

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. For more information, see Cross account resource access in IAM 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. Include actions in a policy to grant permissions to perform the associated operation.

Actions in the ec2 namespace

Some actions for AWS PrivateLink are part of the Amazon EC2 API. These policy actions use the `ec2` prefix. For more information, see AWS PrivateLink actions in the *Amazon EC2 API Reference*.

Actions in the vpce namespace

AWS PrivateLink also provides the `AllowMultiRegion` permissions-only action. This policy action uses the `vpce` prefix.

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. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

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

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

The following condition keys are specific to AWS PrivateLink:

- `ec2:VpceMultiRegion`
- `ec2:VpceServiceName`
- `ec2:VpceServiceOwner`
- `ec2:VpceServicePrivateDnsName`
- `ec2:VpceServiceRegion`
- `ec2:VpceSupportedRegion`

For more information, see Condition keys for 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 called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

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

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

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

Using temporary credentials with AWS PrivateLink

Supports temporary credentials: Yes

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

Cross-service principal permissions for AWS PrivateLink

Supports forward access sessions (FAS): Yes

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

Service roles for AWS 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 Create 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. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

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

For details about actions and resource types defined by AWS 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.

JSON

```
{
    "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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:us-east-1:111111111111:vpc/*",
```

```
                "arn:aws:ec2:us-east-1:111111111111:security-group/*",
                "arn:aws:ec2:us-east-1:111111111111:subnet/*",
                "arn:aws:ec2:us-east-1:111111111111:route-table/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:us-east-1:111111111111: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.

JSON

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

```
                "ec2:CreateVpcEndpointServiceConfiguration"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:111111111111: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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:us-east-1:111111111111:vpc/*",
                "arn:aws:ec2:us-east-1:111111111111:security-group/*",
                "arn:aws:ec2:us-east-1:111111111111:subnet/*",
                "arn:aws:ec2:us-east-1:111111111111:route-table/*"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": "ec2:CreateVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:us-east-1:111111111111:vpc-endpoint/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:VpceServiceName": [
                        "com.amazonaws.111111111111.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.
- You can't use wildcard characters (* or ?) or numeric condition operators with global context keys that reference system-generated identifiers (for example, `aws:PrincipalAccount` or `aws:SourceVpc`).
- When you use a string condition operator, you must use at least six consecutive characters before or after each wildcard character.
- When you specify an ARN in a resource or condition element, the account portion of the ARN can include an account ID or a wildcard character, but not both.
- After you update an endpoint policy, it can take a few minutes for the changes to take effect.

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)

AWS managed policies for AWS PrivateLink

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWS PrivateLink updates to AWS managed policies

View details about updates to AWS managed policies for AWS PrivateLink since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the AWS PrivateLink Document history page.

Change	Description	Date
AWS PrivateLink started tracking changes	AWS PrivateLink started tracking changes for its AWS managed policies.	March 1, 2021

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 or for endpoint service consumers that use cross-Region access. 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
`ActiveConnections`	The number of concurrent active connections. This includes connections in the SYN_SENT and ESTABLISHED states. **Reporting criteria**: The endpoint received traffic during the one-minute period.

Metric	Description
	Statistics: The most useful statistics are `Average`, `Maximum`, and `Minimum`. **Dimensions** • `Endpoint Type`, `Service Name`, `VPC Endpoint Id`, `VPC Id` • `Endpoint Type`, `Service Name`, `Subnet Id`, `VPC Endpoint Id`, `VPC Id`
`BytesProcessed`	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. **Reporting criteria**: The endpoint received traffic during the one-minute period. **Statistics**: The most useful statistics are `Average`, `Sum`, `Maximum`, and `Minimum`. **Dimensions** • `Endpoint Type`, `Service Name`, `VPC Endpoint Id`, `VPC Id` • `Endpoint Type`, `Service Name`, `Subnet Id`, `VPC Endpoint Id`, `VPC Id`

Metric	Description
NewConnections	The number of new connections established through the endpoint. **Reporting criteria**: The endpoint received traffic during the one-minute period. **Statistics**: The most useful statistics are Average, Sum, Maximum, and Minimum. **Dimensions** • Endpoint Type, Service Name, VPC Endpoint Id, VPC Id • Endpoint Type, Service Name, Subnet Id, VPC Endpoint Id, VPC Id
PacketsDropped	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. **Reporting criteria**: The endpoint received traffic during the one-minute period. **Statistics**: The most useful statistics are Average, Sum, and Maximum. **Dimensions** • Endpoint Type, Service Name, VPC Endpoint Id, VPC Id • Endpoint Type, Service Name, Subnet Id, VPC Endpoint Id, VPC Id

Metric	Description
RstPacketsReceived	The number of RST packets received by the endpoint. Increasing values could indicate that the endpoint service is unhealthy. **Reporting criteria**: The endpoint received traffic during the one-minute period. **Statistics**: The most useful statistics are Average, Sum, and Maximum. **Dimensions** • 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	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. **Reporting criteria**: An endpoint connected to the endpoint service sent traffic during the one-minute period. **Statistics**: The most useful statistics are Average and Maximum. **Dimensions** • Service Id • Az, Service Id • Load Balancer Arn, Service Id • Az, Load Balancer Arn, Service Id • Service Id, VPC Endpoint Id
BytesProcessed	The number of bytes exchanged between endpoint services and endpoints, in both directions. **Reporting criteria**: An endpoint connected to the endpoint service sent traffic during the one-minute period. **Statistics**: The most useful statistics are Average, Sum, and Maximum. **Dimensions** • Service Id • Az, Service Id • Load Balancer Arn, Service Id • Az, Load Balancer Arn, Service Id • Service Id, VPC Endpoint Id
EndpointsCount	The number of endpoints connected to the endpoint service.

Metric	Description
	Reporting criteria: There is a nonzero value during the five-minute period. **Statistics**: The most useful statistics are `Average` and `Maximum`. **Dimensions** • `Service Id`
`NewConnections`	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. **Reporting criteria**: An endpoint connected to the endpoint service sent traffic during the one-minute period. **Statistics**: The most useful statistics are `Average`, `Sum`, and `Maximum`. **Dimensions** • `Service Id` • `Az`, `Service Id` • `Load Balancer Arn`, `Service Id` • `Az`, `Load Balancer Arn`, `Service Id` • `Service Id`, `VPC Endpoint Id`

Metric	Description
RstPacketsSent	The number of RST packets sent to endpoints by the endpoint service. Increasing values could indicate that there are unhealthy targets. **Reporting criteria**: An endpoint connected to the endpoint service sent traffic during the one-minute period. **Statistics**: The most useful statistics are Average, Sum, and Maximum. **Dimensions** • 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

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, and other quotas cannot be increased. If you request a quota increase that applies per resource, we increase the quota for all resources in the Region.

To request a quota increase, see Requesting a quota increase in the *Service Quotas User Guide*.

Request throttling

The API actions for AWS PrivateLink are part of the Amazon EC2 API. Amazon EC2 throttles its API requests at the AWS account level. For more information, see Request throttling in the *Amazon EC2 Developer Guide*. In addition, API requests are also throttled at the organization level to help the performance of AWS PrivateLink. If you are using AWS Organizations and you receive a `RequestLimitExceeded` error code while you are still within your account-level API limits, see How to identify AWS accounts that make a large number of API calls. If you need help, contact your account team or open a technical support case using the **VPC** service and the **VPC Endpoints** category. Be sure to attach an image of the `RequestLimitExceeded` error code.

VPC endpoint quotas

Your AWS account has the following quotas related to VPC endpoints.

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
Resource VPC endpoints per VPC	200	Yes	
Service network VPC endpoints per VPC	50	Yes	

Name	Default	Adjustable	Comments
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
Access resources and service networks	AWS PrivateLink supports accessing resources and service networks across VPC and account boundaries.	December 1, 2024
Cross-Region access	A service provider can host a service in one Region and make it available in a set of AWS Regions. A service consumer selects a service Regions when creating an endpoint.	November 26, 2024
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 Insights rules to identify specific endpoints that are the top contributors to the	August 18, 2022

	CloudWatch metrics for AWS PrivateLink.	
IPv6 support	Service providers can enable their endpoint service to accept IPv6 requests, eve n if their backend services support only IPv4. If an endpoint service accepts IPv6 requests, service consumers can enable IPv6 support for their interface endpoints so that they can access the endpoint service over IPv6.	May 11, 2022
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 serv ices.	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
VPC endpoint services	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
Interface VPC endpoints for AWS services	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
VPC endpoints for DynamoDB	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
VPC endpoints for Amazon S3	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