Amazon Virtual Private Cloud

AWS PrivateLink

Amazon Virtual Private Cloud: AWS PrivateLink

Copyright © 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

AWS PrivateLink and VPC endpoints

A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by AWS PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.

VPC endpoints concepts

The following are the key concepts for VPC endpoints:

- **VPC endpoint** — The entry point in your VPC that enables you to connect privately to a service. The following are the different types of VPC endpoints. You create the type of VPC endpoint required by the supported service.
  - Gateway endpoint (p. 21)
  - Interface endpoint (p. 3)
  - Gateway Load Balancer endpoint (p. 18)
- **Endpoint service** — Your own application or service in your VPC. Other AWS principals can create an endpoint from their VPC to your endpoint service.
- **AWS PrivateLink** — A technology that provides private connectivity between VPCs and 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 VPC endpoints.
- **AWS Command Line Interface (AWS CLI)** — Provides commands for a broad set of AWS services, including Amazon VPC. The AWS CLI is supported on Windows, macOS, and Linux. For more information, see AWS Command Line Interface.
- **AWS SDKs** — Provide language-specific APIs. The SDKs take care of many of the connection details, such as calculating signatures, handling request retries, and handling errors. For more information, see AWS SDKs.
- **Query API** — Provides low-level API actions that you call using HTTPS requests. Using the Query API is the most direct way to access Amazon VPC. However, it requires that your application handle low-level details such as generating the hash to sign the request and handling errors. For more information, see the Amazon EC2 API Reference.

Example endpoint configurations

For information about AWS PrivateLink and VPC peering examples, see Examples: Services using AWS PrivateLink and VPC peering in the *Amazon VPC User Guide*.

Pricing for endpoints

For information about endpoint pricing, see AWS PrivateLink Pricing. You can view the total number of endpoints using the Amazon VPC Console, or the AWS CLI.

VPC endpoints

A VPC endpoint enables private connections between your VPC and supported AWS services and VPC endpoint services powered by AWS PrivateLink. AWS PrivateLink is a technology that enables you to privately access services by using private IP addresses. Traffic between your VPC and the other service does not leave the Amazon network. A VPC endpoint does not require an internet gateway, virtual private gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service.

VPC endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components. They allow communication between instances in your VPC and services without imposing availability risks.

The following are the different types of VPC endpoints. You create the type of VPC endpoint that's required by the supported service.

Interface endpoints

An interface endpoint (p. 3) is an elastic network interface with a private IP address from the IP address range of your subnet. It serves as an entry point for traffic destined to a supported AWS service or a VPC endpoint service. Interface endpoints are powered by AWS PrivateLink.

For information about the AWS services that integrate with AWS PrivateLink, see *Services that support AWS PrivateLink* (p. 67). You can also view all of the available AWS service names. For more information, see View available AWS service names (p. 9).

Gateway Load Balancer endpoints

A Gateway Load Balancer endpoint (p. 18) is an elastic network interface with a private IP address from the IP address range of your subnet. Gateway Load Balancer endpoints are powered by AWS PrivateLink. This type of endpoint serves as an entry point to intercept traffic and route it to a service that you've configured using Gateway Load Balancers, for example, for security inspection. You specify a Gateway Load Balancer endpoint as a target for a route in a route table. Gateway Load Balancer endpoints are supported for endpoint services that are configured for Gateway Load Balancers only.

Gateway endpoints

A gateway endpoint (p. 21) is for the following supported AWS services:

- Amazon S3
- DynamoDB

You specify a gateway endpoint as a route table target for traffic that is destined for the supported AWS services.

Interface VPC endpoints (AWS PrivateLink)

An interface VPC endpoint (interface endpoint) allows you to connect to services powered by AWS PrivateLink. These services include some AWS services, services hosted by other AWS customers and Partners in their own VPCs (referred to as *endpoint services*), and supported AWS Marketplace Partner

services. The owner of the service is the *service provider*, and you, as the principal creating the interface endpoint, are the *service consumer*.

The following are the general steps for setting up an interface endpoint:

1. Choose the VPC in which to create the interface endpoint, and provide the name of the AWS service, endpoint service, or AWS Marketplace service to which you're connecting.
2. Choose a subnet in your VPC to use the interface endpoint. We create an *endpoint network interface* in the subnet. An endpoint network interface is assigned a private IP address from the IP address range of your subnet, and keeps this IP address until the interface endpoint is deleted. You can specify more than one subnet in different Availability Zones (as supported by the service) to help ensure that your interface endpoint is resilient to Availability Zone failures. In that case, we create an endpoint network interface in each subnet that you specify.

   Note
   An endpoint network interface is a requester-managed network interface. You can view it in your account, but you cannot manage it yourself. For more information, see Requester-managed network interfaces.
3. Specify the security groups to associate with the endpoint network interface. The security group rules control the traffic to the endpoint network interface from resources in your VPC. If you do not specify a security group, we associate the default security group for the VPC.
4. (Optional, AWS services and AWS Marketplace Partner services only) Enable private DNS (p. 5) for the endpoint so you can make requests to the service using its default DNS hostname.

   Important
   Private DNS is turned on by default for endpoints created for AWS services and AWS Marketplace Partner services.
   Private DNS is turned on in the other subnets which are in the same VPC and Availability Zone or Local Zone.
5. When the service provider and the consumer are in different accounts, see the section called “Interface endpoint Availability Zone considerations” (p. 8) for information about how to use Availability Zone IDs to identify the interface endpoint Availability Zone.
6. After you create the interface endpoint, it's available to use when it's accepted by the service provider. The service provider must configure the service to accept requests automatically or manually. AWS services and AWS Marketplace services generally accept all endpoint requests automatically. For more information about the lifecycle of an endpoint, see Interface endpoint lifecycle (p. 8).

Services cannot initiate requests to resources in your VPC through the endpoint. An endpoint only returns responses to traffic that is initiated from resources in your VPC. Before you integrate a service and an endpoint, review the service-specific VPC endpoint documentation for any service-specific configuration and limitations.

Contents

Private DNS for interface endpoints

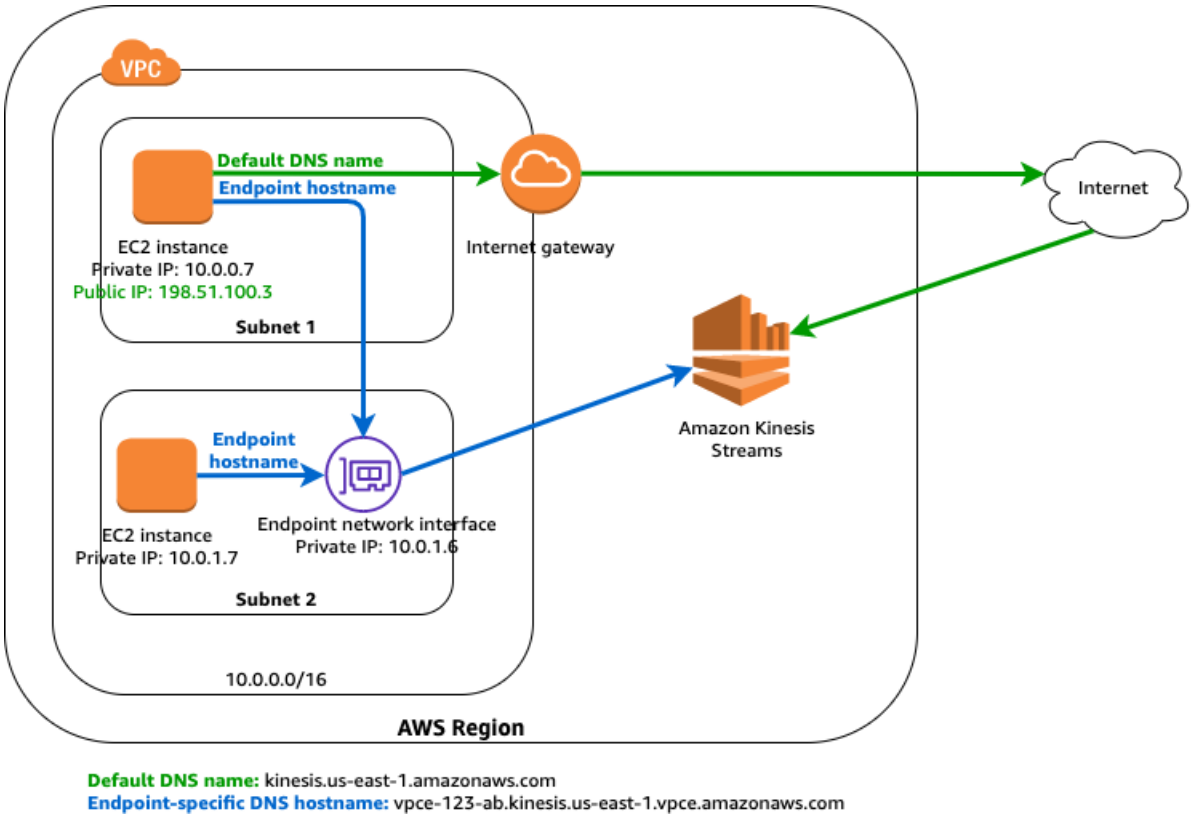
Important
Private DNS is not supported for Amazon S3 interface endpoints.

When you create an interface endpoint, we generate endpoint-specific DNS hostnames that you can use to communicate with the service. For AWS services and AWS Marketplace Partner services, the private DNS option (turned on by default) associates a private hosted zone with your VPC. The hosted zone contains a record set for the default DNS name for the service (for example, `ec2.us-east-1.amazonaws.com`) that resolves to the private IP addresses of the endpoint network interfaces in your VPC. This allows you to make requests to the service using its default DNS hostname instead of the endpoint-specific DNS hostnames. For example, if your existing applications make requests to an AWS service, they can continue to make requests through the interface endpoint without requiring any configuration changes.

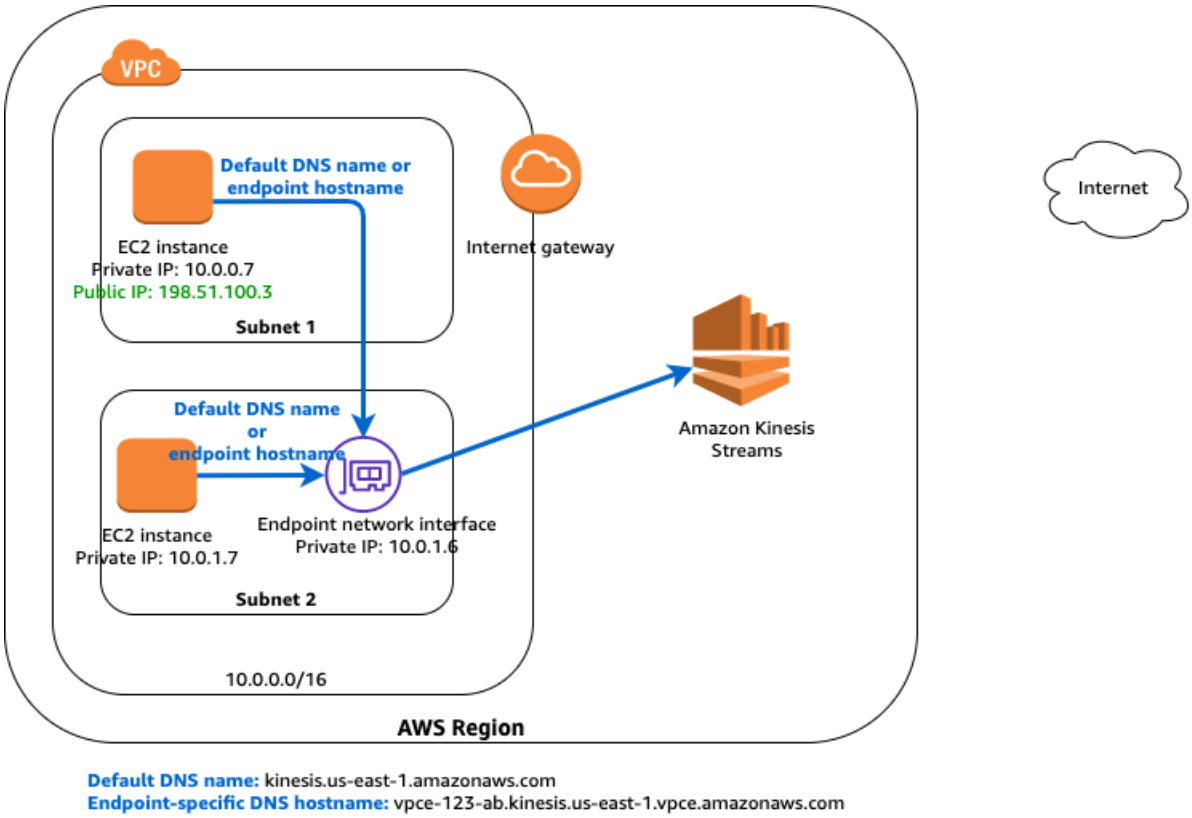
In the example shown in the following diagram, there is an interface endpoint for Amazon Kinesis Data Streams and an endpoint network interface in subnet 2. Private DNS for the interface endpoint is turned off. The route tables for the subnets have the following routes.

Subnet 1	
Destination	**Target**
10.0.0.0/16	Local
0.0.0.0/0	*internet-gateway-id*
Subnet 2	
Destination	**Target**
10.0.0.0/16	Local

Instances in either subnet can send requests to Amazon Kinesis Data Streams through the interface endpoint using an endpoint-specific DNS hostname. Instances in subnet 1 can communicate with Amazon Kinesis Data Streams over public IP address space in the AWS Region using its default DNS name.



In the next diagram, private DNS for the endpoint is turned on. Instances in either subnet can send requests to Amazon Kinesis Data Streams through the interface endpoint using either the default DNS hostname or the endpoint-specific DNS hostname.



Important
To use private DNS, you must set the following VPC attributes to `true`: `enableDnsHostnames` and `enableDnsSupport`. For more information, see Viewing and updating DNS support for your VPC. IAM users must have permission to work with hosted zones. For more information, see Authentication and Access Control for Route 53.

Interface endpoint properties and limitations

To use interface endpoints, you need to be aware of their properties and current limitations:

- For each interface endpoint, you can choose only one subnet per Availability Zone.
- Services might not be available in all Availability Zones through an interface endpoint. To find out which Availability Zones are supported, use the describe-vpc-endpoint-services command or use the Amazon VPC console. For more information, see Create an interface endpoint (p. 10).
- When you create an interface endpoint, the endpoint is created in the Availability Zone that is mapped to your account and that is independent from other accounts. When the service provider and the consumer are in different accounts, see the section called "Interface endpoint Availability Zone considerations" (p. 8) for information about how to use Availability Zone IDs to identify the interface endpoint Availability Zone.
- When the service provider and the consumer have different accounts and use multiple Availability Zones, and the consumer views the VPC endpoint service information, the response only includes the common Availability Zones. For example, when the service provider account uses `us-east-1a` and `us-east-1c` and the consumer uses `us-east-1a` and `us-east-1b`, the response includes the VPC endpoint services in the common Availability Zone, `us-east-1a`.
- By default, each interface endpoint can support a bandwidth of up to 10 Gbps per Availability Zone. and bursts of up to 40Gbps. If your application needs higher bursts or sustained throughput, contact AWS support.

- If the network ACL for your subnet restricts traffic, you might not be able to send traffic through the endpoint network interface. Ensure that you add appropriate rules that allow traffic to and from the CIDR block of the subnet.
- Ensure that the security group that's associated with the endpoint network interface allows communication between the endpoint network interface and the resources in your VPC that communicate with the service. To ensure that command line tools such as the AWS CLI can make requests over HTTPS from resources in the VPC to an AWS service, the security group must allow inbound HTTPS (port 443) traffic .
- An interface endpoint supports TCP traffic only.
- When you create an endpoint, you can attach an endpoint policy to it that controls access to the service to which you are connecting. For more information, see Policy Best Practices and the section called “Control access to services” (p. 37).
- Review the service-specific limits for your endpoint service.
- Endpoints are supported within the same Region only. You cannot create an endpoint between a VPC and a service in a different Region.
- Endpoints support IPv4 traffic only.
- You cannot transfer an endpoint from one VPC to another, or from one service to another.
- You have a quota on the number of endpoints you can create per VPC. For more information, see AWS PrivateLink quotas (p. 72).

Connection to on-premises data centers

You can use the following types of connections for a connection between an interface endpoint and your on-premises data center:

- AWS Direct Connect
- AWS Site-to-Site VPN

Interface endpoint lifecycle

An interface endpoint goes through various stages starting from when you create it (the endpoint connection request). At each stage, there might be actions that the service consumer and service provider can take.

The following rules apply:

- A service provider can configure their service to accept interface endpoint requests automatically or manually. AWS services and AWS Marketplace services generally accept all endpoint requests automatically.
- A service provider cannot delete an interface endpoint to their service. Only the service consumer that requested the interface endpoint connection can delete the interface endpoint.
- A service provider can reject the interface endpoint after it has been accepted (either manually or automatically) and is in the `available` state.

Interface endpoint Availability Zone considerations

When you create an interface endpoint, the endpoint is created in the Availability Zone that is mapped to your account and that is independent from other accounts. When the service provider and the consumer

are in different accounts, use the Availability Zone ID to uniquely and consistently identify the interface endpoint Availability Zone. For example, `use1-az1` is an Availability Zone ID for the `us-east-1` Region and maps to the same location in every AWS account. For information about Availability Zone IDs, see AZ IDs for Your Resources in the *AWS RAM User Guide* or use describe-availability-zones.

Services might not be available in all Availability Zones through an interface endpoint. You can use any of the following operations to find out which Availability Zones are supported for a service:

- describe-vpc-endpoint-services (AWS CLI)
- DescribeVpcEndpointServices (API)
- The Amazon VPC console when you create an interface endpoint. For more information, see the section called “Create an interface endpoint” (p. 10).

View available AWS service names

When you use the Amazon VPC console to create an endpoint, you can get a list of available AWS service names.

When you use the AWS CLI to create an endpoint, you can use the describe-vpc-endpoint-services command to view the service names, and then create the endpoint using the create-vpc-endpoint command .

Console

To view available AWS services using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints**, **Create Endpoint**.
3. In the **Service Name** section, the available services are listed.

Command line

To view available AWS services using the AWS CLI

- Use the describe-vpc-endpoint-services command to get a list of available services. In the output that's returned, take note of the name of the service to which to connect. The `ServiceType` field indicates whether you connect to the service via an interface or gateway endpoint. The `ServiceName` field provides the name of the service.

```
aws ec2 describe-vpc-endpoint-services
```

```
{
    "VpcEndpoints": [
        {
            "VpcEndpointId": "vpce-08a979e28f97a9f7c",
            "VpcEndpointType": "Interface",
            "VpcId": "vpc-06e4ab6c6c3b23ae3",
            "ServiceName": "com.amazonaws.us-east-2.monitoring",
            "State": "available",
            "PolicyDocument": "{\n  \"Statement\": [\n    {\n      \"Action\": \"*
\", \n      \"Effect\": \"Allow\", \n      \"Principal\": \"*\", \n      \"Resource
\": \"*\"\n    }\n  ]\n}",
            "RouteTableIds": [],
            "SubnetIds": [
                "subnet-0931fc2fa5f1cbe44"
            ],
```

```
            "Groups": [
                {
                    "GroupId": "sg-06e1d57ab87d8f182",
                    "GroupName": "default"
                }
            ],
            "PrivateDnsEnabled": false,
            "RequesterManaged": false,
            "NetworkInterfaceIds": [
                "eni-019b0bb3ede80ebfd"
            ],
            "DnsEntries": [
                {
                    "DnsName": "vpce-08a979e28f97a9f7c-4r5zme9n.monitoring.us-
east-2.vpce.amazonaws.com",
                    "HostedZoneId": "ZC8PG0KIFKBRI"
                },
                {
                    "DnsName": "vpce-08a979e28f97a9f7c-4r5zme9n-us-
east-2c.monitoring.us-east-2.vpce.amazonaws.com",
                    "HostedZoneId": "ZC8PG0KIFKBRI"
                }
            ],
            "CreationTimestamp": "2019-06-04T19:10:37.000Z",
            "Tags": [],
            "OwnerId": "123456789012"
        }
    ]
```


To view available AWS services using the AWS Tools for Windows PowerShell

- Get-EC2VpcEndpointService

To view available AWS services using the API

- DescribeVpcEndpointServices

Create an interface endpoint

To create an interface endpoint, you must specify the VPC in which to create the interface endpoint, and the service to which to establish the connection.

For AWS services, or AWS Marketplace Partner services, you can optionally turn on private DNS (p. 5) for the endpoint so that you can make requests to the service using its default DNS hostname.

Important
Private DNS is turned on by default for endpoints created for AWS services and AWS Marketplace Partner services.

Console

To create an interface endpoint to an AWS service using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints**, **Create Endpoint**.
3. For **Service category**, ensure that **AWS services** is selected.
4. For **Service Name**, choose the service to which to connect. For **Type**, ensure that it indicates **Interface**.

5. Complete the following information and then choose **Create endpoint**.

   - For **VPC**, select a VPC in which to create the endpoint.
   - For **Subnets**, select the subnets (Availability Zones) in which to create the endpoint network interfaces.

     Not all Availability Zones may be supported for all AWS services.
   - To turn on private DNS for the interface endpoint, for **Enable DNS Name**, select the check box.

     Important
     Private DNS is not supported for Amazon S3 interface endpoints.

     This option is tuned on by default. To use the private DNS option, the following attributes of your VPC must be set to `true`: `enableDnsHostnames` and `enableDnsSupport`. For more information, see Viewing and updating DNS support for your VPC.
   - For **Security group**, select the security groups to associate with the endpoint network interfaces.
   - (Optional) Add or remove a tag.

     [Add a tag] Choose **Add tag** and do the following:

     - For **Key**, enter the key name.
     - For **Value**, enter the key value.

     [Remove a tag] Choose the delete button ("x") to the right of the tag's Key and Value.

To create an interface endpoint to an endpoint service, you must have the name of the service to which to connect. The service provider can provide you with the name.

To create an interface endpoint to an endpoint service

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints**, **Create Endpoint**.
3. For **Service category**, choose **Find service by name**.
4. For **Service Name**, enter the name of the service (for example, `com.amazonaws.vpce.us-east-1.vpce-svc-0e123abc123198abc`) and choose **Verify**.
5. Complete the following information and then choose **Create endpoint**.

   - For **VPC**, select a VPC in which to create the endpoint.
   - For **Subnets**, select the subnets (Availability Zones) in which to create the endpoint network interfaces.

     Not all Availability Zones may be supported for the service.
   - For **Security group**, select the security groups to associate with the endpoint network interfaces.
   - (Optional) Add or remove a tag.

     [Add a tag] Choose **Add tag** and do the following:

     - For **Key**, enter the key name.
     - For **Value**, enter the key value.

     [Remove a tag] Choose the delete button ("x") to the right of the tag's Key and Value.

To create an interface endpoint to an AWS Marketplace partner service

1. Go to the PrivateLink page in AWS Marketplace and subscribe to a service from a software as a service (SaaS) provider. Services that support interface endpoints include an option to connect via an endpoint.
2. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
3. In the navigation pane, choose **Endpoints**, **Create Endpoint**.
4. For **Service category**, choose **Your AWS Marketplace services**.
5. Choose the AWS Marketplace service to which you've subscribed.
6. Complete the following information and then choose **Create endpoint**.

   - For **VPC**, select a VPC in which to create the endpoint.
   - For **Subnets**, select the subnets (Availability Zones) in which to create the endpoint network interfaces.

     Not all Availability Zones may be supported for the service.
   - For **Security group**, select the security groups to associate with the endpoint network interfaces.
   - (Optional) Add or remove a tag.

     [Add a tag] Choose **Add tag** and do the following:
     - For **Key**, enter the key name.
     - For **Value**, enter the key value.

     [Remove a tag] Choose the delete button ("x") to the right of the tag's Key and Value.

Command line

To create an interface endpoint using the AWS CLI

1. Use the describe-vpc-endpoint-services command to get a list of available services. In the output that's returned, take note of the name of the service to which to connect. The `ServiceType` field indicates whether you connect to the service via an interface or gateway endpoint. The `ServiceName` field provides the name of the service.
2. To create an interface endpoint, use the create-vpc-endpoint command and specify the VPC ID, type of VPC endpoint (interface), service name, subnets that will use the endpoint, and security groups to associate with the endpoint network interfaces.

   The following example creates an interface endpoint to the Elastic Load Balancing service.

```
aws ec2 create-vpc-endpoint --vpc-id vpc-ec43eb89 --vpc-endpoint-type Interface
 --service-name com.amazonaws.us-east-1.elasticloadbalancing --subnet-id subnet-
abababab --security-group-id sg-1a2b3c4d
```

```
{
    "VpcEndpoint": {
        "PolicyDocument": "{\n  \"Statement\": [\n    {\n      \"Action\": \"*\",
 \n      \"Effect\": \"Allow\", \n      \"Principal\": \"*\", \n      \"Resource\":
 \"*\"\n    }\n  ]\n}",
        "VpcId": "vpc-ec43eb89",
        "NetworkInterfaceIds": [
            "eni-bf8aa46b"
        ],
        "SubnetIds": [
            "subnet-abababab"
```

```
        ],
        "PrivateDnsEnabled": true,
        "State": "pending",
        "ServiceName": "com.amazonaws.us-east-1.elasticloadbalancing",
        "RouteTableIds": [],
        "Groups": [
            {
                "GroupName": "default",
                "GroupId": "sg-1a2b3c4d"
            }
        ],
        "VpcEndpointId": "vpce-088d25a4bbf4a7abc",
        "VpcEndpointType": "Interface",
        "CreationTimestamp": "2017-09-05T20:14:41.240Z",
        "DnsEntries": [
            {
                "HostedZoneId": "Z7HUB22UULQXV",
                "DnsName": "vpce-088d25a4bbf4a7abc-
ks83awe7.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
            },
            {
                "HostedZoneId": "Z7HUB22UULQXV",
                "DnsName": "vpce-088d25a4bbf4a7abc-ks83awe7-us-
east-1a.elasticloadbalancing.us-east-1.vpce.amazonaws.com"
            },
            {
                "HostedZoneId": "Z1K56Z6FNPJRR",
                "DnsName": "elasticloadbalancing.us-east-1.amazonaws.com"
            }
        ]
    }
}
```

Alternatively, the following example creates an interface endpoint to an endpoint service in another account (the service provider provides you with the name of the endpoint service).

```
aws ec2 create-vpc-endpoint --vpc-id vpc-ec43eb89 --vpc-endpoint-type Interface
 --service-name com.amazonaws.vpce.us-east-1.vpce-svc-0e123abc123198abc --subnet-
id subnet-abababab --security-group-id sg-1a2b3c4d
```

In the output that's returned, take note of the `privateDnsNames` fields. You can use these DNS names to access the AWS service.

To describe available services and create a VPC endpoint using the AWS Tools for Windows PowerShell

- Get-EC2VpcEndpointService
- New-EC2VpcEndpoint

To describe available services and create a VPC endpoint using the API

- DescribeVpcEndpointServices
- CreateVpcEndpoint

View your interface endpoint

After you've created an interface endpoint, you can view information about it.

Console

To view information about 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** and select your interface endpoint.
3. To view information about the interface endpoint, choose **Details**. The **DNS Names** field displays the DNS names to use to access the service.
4. To view the subnets in which the interface endpoint has been created, and the ID of the endpoint network interface in each subnet, choose **Subnets**.
5. To view the security groups that are associated with the endpoint network interface, choose **Security Groups**.

Command line

To describe your interface endpoint using the AWS CLI

- You can describe your endpoint using the describe-vpc-endpoints command.

```
aws ec2 describe-vpc-endpoints --vpc-endpoint-ids vpce-088d25a4bbf4a7abc
```

To describe your VPC endpoints using the AWS Tools for PowerShell or API

- Get-EC2VpcEndpoint (Tools for Windows PowerShell)
- DescribeVpcEndpoints (Amazon EC2 Query API)

Create and manage a notification for an interface endpoint

You can create a notification to receive alerts for specific events that occur on your interface endpoint. For example, you can receive an email when the interface endpoint is accepted by the service provider. To create a notification, you must associate an Amazon SNS topic with the notification. You can subscribe to the SNS topic to receive an email notification when an endpoint event occurs.

The Amazon SNS topic that you use for notifications must have a topic policy that allows Amazon's VPC endpoint service to publish notifications on your behalf. Ensure that you include the following statement in your topic policy. For more information, see Identity and Access Management in Amazon SNS in the *Amazon Simple Notification Service Developer Guide*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vpce.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:region:account:topic-name"
    }
  ]
}
```

Command line

To create and manage a notification using the AWS CLI

1. To create a notification for an interface endpoint, use the create-vpc-endpoint-connection-notification command. Specify the ARN of the SNS topic, the events for which to be notified, and the ID of the endpoint, as shown in the following example.

```
aws ec2 create-vpc-endpoint-connection-notification --connection-notification-
arn arn:aws:sns:us-east-2:123456789012:EndpointNotification --connection-events
 Accept Reject --vpc-endpoint-id vpce-123abc3420c1931d7
```

2. To view your notifications, use the describe-vpc-endpoint-connection-notifications command.

```
aws ec2 describe-vpc-endpoint-connection-notifications
```

3. To change the SNS topic or endpoint events for the notification, use the modify-vpc-endpoint-connection-notification command.

```
aws ec2 modify-vpc-endpoint-connection-notification --connection-notification-
id vpce-nfn-008776de7e03f5abc --connection-events Accept --connection-notification-
arn arn:aws:sns:us-east-2:123456789012:mytopic
```

4. To delete a notification, use the delete-vpc-endpoint-connection-notifications command.

```
aws ec2 delete-vpc-endpoint-connection-notifications --connection-notification-
ids vpce-nfn-008776de7e03f5abc
```

Access a service through an interface endpoint

After you've created an interface endpoint, you can submit requests to the supported service via an endpoint URL. You can use the following:

- If you have turned on private DNS for the endpoint (a private hosted zone; applicable to AWS services and AWS Marketplace Partner services only), the default DNS hostname for the AWS service for the Region. For example, `ec2.us-east-1.amazonaws.com`.

  Important
  Private DNS is not supported for Amazon S3 interface endpoints.
- The endpoint-specific Regional DNS hostname that we generate for the interface endpoint. The hostname includes a unique endpoint identifier, service identifier, the Region, and `vpce.amazonaws.com` in its name. For example, `vpce-0fe5b17a0707d6abc-29p5708s.ec2.us-east-1.vpce.amazonaws.com`.
- The endpoint-specific zonal DNS hostname that we generate for each Availability Zone in which the endpoint is available. The hostname includes the Availability Zone in its name. For example, `vpce-0fe5b17a0707d6abc-29p5708s-us-east-1a.ec2.us-east-1.vpce.amazonaws.com`. You might use this option if your architecture isolates Availability Zones (for example, for fault containment or to reduce Regional data transfer costs).

  A request to the zonal DNS hostname is destined to the corresponding Availability Zone location in the service provider's account, which might not have the same Availability Zone name as your account. For more information, see Region and Availability Zone Concepts.
- The private IP address of the endpoint network interface in the VPC.

To get the Regional and zonal DNS names, see View your interface endpoint (p. 13).

For example, in a subnet in which you have an interface endpoint to Elastic Load Balancing and for which you have not turned on the private DNS option, use the following AWS CLI command from an instance to describe your load balancers. The command uses the endpoint-specific Regional DNS hostname to make the request using the interface endpoint.

```
aws elbv2 describe-load-balancers --endpoint-url https://vpce-0f89a33420c193abc-bluzidnv.elasticloadbalancing.us-east-1.vpce.amazonaws.com/
```

If you turn on the private DNS option, you do not have to specify the endpoint URL in the request. The AWS CLI uses the default endpoint for the AWS service for the Region (`elasticloadbalancing.us-east-1.amazonaws.com`).

Modify an interface endpoint

You can modify the following attributes of an interface endpoint:

- The subnet in which the interface endpoint is located
- The security groups that are associated with the endpoint network interface
- The tags
- The private DNS option

  Note
  When you turn on private DNS, it might take a few minutes for the private IP addresses to become available.
- The endpoint policy (if supported by the service)

If you remove a subnet for the interface endpoint, the corresponding endpoint network interface in the subnet is deleted.

Console

To change the subnets for an interface endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints** and select the interface endpoint.
3. Choose **Actions**, **Manage Subnets**.
4. Select or deselect the subnets as required, and choose **Modify Subnets**.

To add or remove the security groups associated with an interface endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints** and select the interface endpoint.
3. Choose **Actions**, **Manage security groups**.
4. Select or deselect the security groups as required, and choose **Save**.

To add or remove an interface endpoint tag

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 and choose **Actions**, **Add/Edit Tags**.

4. Add or remove a tag.

   [Add a tag] Choose **Create tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Choose the delete button ("x") to the right of the tag's Key and Value.

To modify the private DNS option

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints** and select the interface endpoint.
3. Choose **Actions**, **Modify Private DNS names**.
4. Set the option as required, and choose **Modify Private DNS names**.

To update the endpoint policy

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

Command line

To modify a VPC endpoint using the AWS CLI

1. Use the describe-vpc-endpoints command to get the ID of your interface endpoint.

```
aws ec2 describe-vpc-endpoints
```

2. The following example uses the modify-vpc-endpoint command to add subnet `subnet-aabb1122` to the interface endpoint.

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-0fe5b17a0707d6abc --add-subnet-id subnet-aabb1122
```

To modify a VPC endpoint using the AWS Tools for Windows PowerShell or an API

- Edit-EC2VpcEndpoint (AWS Tools for Windows PowerShell)
- ModifyVpcEndpoint (Amazon EC2 Query API)

To add or remove a VPC endpoint tag using the AWS Tools for Windows PowerShell or an API

- tag-resource (AWS CLI)
- TagResource (AWS Tools for Windows PowerShell)
- untag-resource (AWS CLI)
- TagResource (AWS Tools for Windows PowerShell)

Gateway Load Balancer endpoints (AWS PrivateLink)

A Gateway Load Balancer endpoint enables you to intercept traffic and route it to a service that you've configured using Gateway Load Balancers, for example, for security inspection. The owner of the service is the *service provider*, and you, as the principal creating the Gateway Load Balancer endpoint, are the *service consumer*.

The following are the general steps for setting up a Gateway Load Balancer endpoint:

1. Ensure that a Gateway Load Balancer endpoint service is configured. For more information, see VPC endpoint services for Gateway Load Balancer endpoints (p. 44).
2. Choose the VPC in which to create the Gateway Load Balancer endpoint, and provide the name of the service.
3. Choose a subnet in your VPC to use the Gateway Load Balancer endpoint. We create an *endpoint network interface* in the subnet. An endpoint network interface is assigned a private IP address from the IP address range of your subnet, and keeps this IP address until the Gateway Load Balancer endpoint is deleted.

   Note
   An endpoint network interface is a requester-managed network interface. You can view it in your account, but you cannot manage it yourself. For more information, see Requester-managed network interfaces.
   You can specify only one subnet for the Gateway Load Balancer endpoint. You cannot change the subnet later.
4. After you create the Gateway Load Balancer endpoint, it's available to use when it's accepted by the service provider. The service provider can configure the service to accept requests automatically or manually.
5. Configure your subnet route table and gateway route table to point traffic to the Gateway Load Balancer endpoint. For more information, see Routing to a Gateway Load Balancer endpoint in the *Amazon VPC User Guide*.

Contents

Gateway Load Balancer endpoint properties and limitations

To use a Gateway Load Balancer endpoint, be aware of the following:

- For each Gateway Load Balancer endpoint, you can choose only one Availability Zone (subnet) in your VPC. You cannot change the subnet later. To use a Gateway Load Balancer endpoint in a different subnet, create a new Gateway Load Balancer endpoint in that subnet. You can create a single Gateway Load Balancer endpoint per Availability Zone for a service.
- Each Gateway Load Balancer endpoint supports a maximum bandwidth of up to 40 Gbps.

- If the network ACL for your subnet restricts traffic, you might not be able to send traffic through the Gateway Load Balancer endpoint. Ensure that you add appropriate rules that allow traffic to and from the CIDR block of the subnet.
- Security groups are not supported.
- Endpoint policies are not supported.
- A service might not be available in all Availability Zones through a Gateway Load Balancer endpoint. To find out which Availability Zones are supported, use the describe-vpc-endpoint-services command or use the Amazon VPC console. For more information, see Create a Gateway Load Balancer endpoint (p. 19).
- When you create a Gateway Load Balancer endpoint, the endpoint is created in the Availability Zone that is mapped to your account and that is independent from other accounts. When the service provider and the consumer are in different accounts, use the Availability Zone ID to uniquely and consistently identify the endpoint Availability Zone. For example, `use1-az1` is an Availability Zone ID for the `us-east-1` Region and maps to the same location in every AWS account. For information about Availability Zone IDs, see AZ IDs for Your Resources in the *AWS RAM User Guide* or use describe-availability-zones.
- To keep traffic within the same Availability Zone, we recommend that you create a Gateway Load Balancer endpoint in each Availability Zone that you will send traffic to.
- Endpoints are supported within the same Region only. You cannot create an endpoint between a VPC and a service in a different Region.
- Endpoints support IPv4 traffic only.
- You cannot transfer an endpoint from one VPC to another, or from one service to another.
- You have a quota on the number of endpoints you can create per VPC. For more information, see AWS PrivateLink quotas (p. 72).

Gateway Load Balancer endpoint lifecycle

A Gateway Load Balancer endpoint goes through various stages, starting from when you create it (the endpoint connection request). At each stage, there might be actions that the service consumer and service provider can take.

The following rules apply:

- A service provider can configure their service to accept Gateway Load Balancer endpoint requests automatically or manually.
- A service provider cannot delete a Gateway Load Balancer endpoint to their service. Only the service consumer that requested the connection can delete the Gateway Load Balancer endpoint.
- A service provider can reject the Gateway Load Balancer endpoint after it has been accepted and is in the `available` state.

Pricing for Gateway Load Balancer endpoints

You are charged for creating and using a Gateway Load Balancer endpoint to a service. Hourly usage rates and data processing rates apply. For more information, see AWS PrivateLink Pricing. You can view the total number of Gateway Load Balancer endpoints using the Amazon VPC Console, or the AWS CLI.

Create a Gateway Load Balancer endpoint

To create a Gateway Load Balancer endpoint, you must specify the VPC in which to create the endpoint, and the service to which to establish the connection.

Console

To create 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**, **Create Endpoint**.
3. For **Service category**, choose **Find service by name**.
4. For **Service Name**, enter the name of the service and choose **Verify**.
5. Complete the following information and then choose **Create endpoint**.
   - For **VPC**, select a VPC in which to create the endpoint.
   - For **Subnets**, select the subnet (Availability Zone) in which to create the Gateway Load Balancer endpoint.
   - (Optional) To add a tag, choose **Add tag** and then specify a key and value for the tag.

Command line

To create a Gateway Load Balancer endpoint using the AWS CLI

Use the create-vpc-endpoint command and specify the VPC ID, type of VPC endpoint (Gateway Load Balancer), service name, and the subnet in which to create the Gateway Load Balancer endpoint.

```
aws ec2 create-vpc-endpoint --vpc-endpoint-type GatewayLoadBalancer --vpc-id vpc-id --
subnet-ids subnet-id --service-name gateway-load-balancer-service-name
```

To create a VPC endpoint using the AWS Tools for Windows PowerShell or API

- New-EC2VpcEndpoint
- CreateVpcEndpoint

View your Gateway Load Balancer endpoint

After you've created a Gateway Load Balancer endpoint, you can view information about it.

Console

To view information about 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** and select your Gateway Load Balancer endpoint.
3. Choose **Details**.
4. To view the subnet in which the Gateway Load Balancer endpoint has been created, and the ID of the endpoint network interface, choose **Subnets**.

Command line

To describe your Gateway Load Balancer endpoint using a command line tool or API

- describe-vpc-endpoints (AWS CLI)
- Get-EC2VpcEndpoint (Tools for Windows PowerShell)
- DescribeVpcEndpoints (Amazon EC2 Query API)

Add or remove tags for a Gateway Load Balancer endpoint

You can add or remove the tags for your Gateway Load Balancer endpoint.

Console

To add or remove a tag

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints**.
3. Select the Gateway Load Balancer endpoint and choose **Actions**, **Add/Edit Tags**.
4. Add or remove a tag.

   [Add a tag] Choose **Create tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Choose the delete button (“x”) to the right of the tag’s Key and Value.

Command line

To add or remove tags using a command line tool or an API

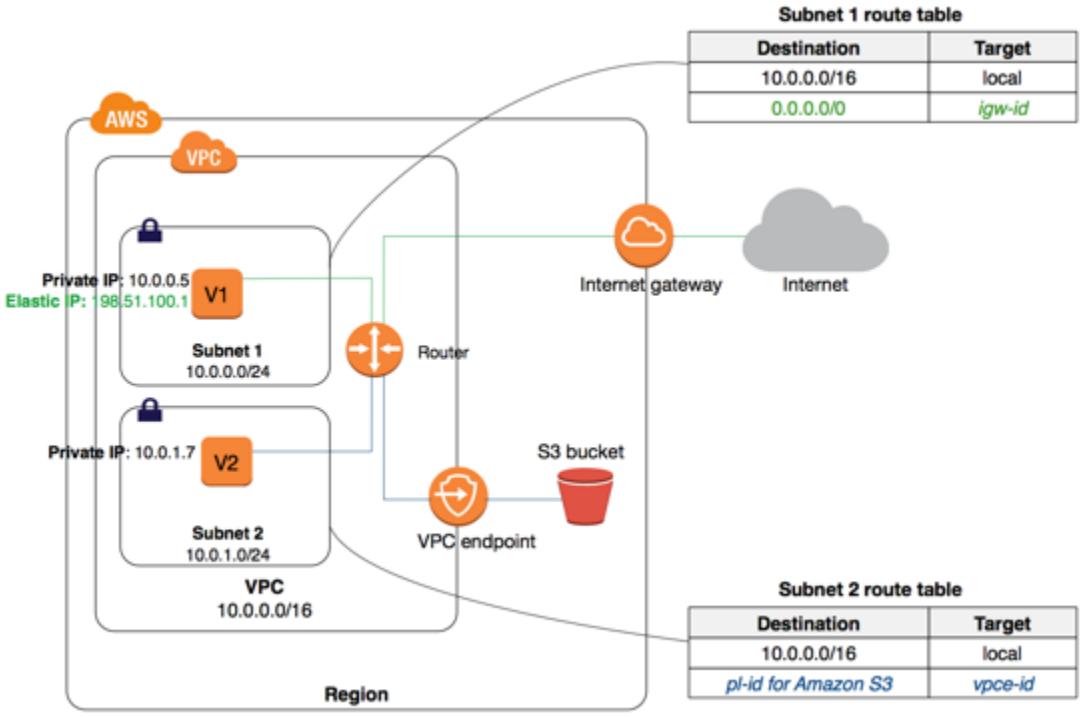
- Use create-tags and delete-tags. (AWS CLI)
- Use New-EC2Tag and Remove-EC2Tag (AWS Tools for Windows PowerShell)
- Use CreateTags and DeleteTags. (Amazon EC2 Query API)

Gateway VPC endpoints

To create and set up a gateway endpoint, follow these general steps:

1. Specify the VPC in which to create the endpoint, and the service to which you're connecting. A service is identified by an AWS managed *prefix list*—the name and ID of a service for a Region. An AWS prefix list ID uses the form `pl-xxxxxxx` and an AWS prefix list name uses the form "com.amazonaws.*region*.*service*". Use the AWS prefix list name (service name) to create an endpoint.
2. Attach an *endpoint policy* to your endpoint that allows access to some or all of the service to which you're connecting. For more information, see Use VPC endpoint policies (p. 37).
3. Specify one or more route tables in which to create routes to the service. Route tables control the routing of traffic between your VPC and the other service. Each subnet that's associated with one of these route tables has access to the endpoint, and traffic from instances in these subnets to the service is then routed through the endpoint.

In the following diagram, instances in subnet 2 can access Amazon S3 through the gateway endpoint.



You can create multiple endpoints in a single VPC, for example, to multiple services. You can also create multiple endpoints for a single service, and use different route tables to enforce different access policies from different subnets to the same service.

After you've created an endpoint, you can modify the endpoint policy that's attached to your endpoint, and add or remove the route tables that are used by the endpoint.

Contents

Pricing for gateway endpoints

There is no additional charge for using gateway endpoints. Standard charges for data transfer and resource usage apply. For more information about pricing, see Amazon EC2 Pricing.

Routing for gateway endpoints

When you create or modify an endpoint, you specify the VPC route tables that are used to access the service via the endpoint. A route is automatically added to each of the route tables with a destination

that specifies the AWS prefix list ID of the service (`pl-xxxxxxxx`), and a target with the endpoint ID (`vpce-xxxxxxxx`); for example:

Destination	Target
10.0.0.0/16	Local
pl-1a2b3c4d	vpce-11bb22cc

The prefix list ID logically represents the range of public IP addresses used by the service. All instances in subnets associated with the specified route tables automatically use the endpoint to access the service. Subnets that are not associated with the specified route tables do not use the endpoint. This allows you to keep resources in other subnets separate from your endpoint.

To view the current public IP address range for a service, you can use the describe-prefix-lists command.

> **Note**
> The range of public IP addresses for a service may change from time to time. Consider the implications before you make routing or other decisions based on the current IP address range for a service.

The following rules apply:

- You can have multiple endpoint routes to different services in a route table, and you can have multiple endpoint routes to the same service in different route tables. But you cannot have multiple endpoint routes to the same service in a single route table. For example, if you create two endpoints to Amazon S3 in your VPC, you cannot create endpoint routes for both endpoints in the same route table.
- You cannot explicitly add, modify, or delete an endpoint route in your route table by using the route table APIs, or by using the Route Tables page in the Amazon VPC console. You can only add an endpoint route by associating a route table with an endpoint. To change the route tables that are associated with your endpoint, you can modify the endpoint (p. 36).
- An endpoint route is automatically deleted when you remove the route table association from the endpoint (by modifying the endpoint), or when you delete your endpoint.

We use the most specific route that matches the traffic to determine how to route the traffic (longest prefix match). If you have an existing route in your route table for all internet traffic (`0.0.0.0/0`) that points to an internet gateway, the endpoint route takes precedence for all traffic destined for the service, because the IP address range for the service is more specific than `0.0.0.0/0`. All other internet traffic goes to your internet gateway, including traffic that's destined for the service in other Regions.

However, if you have existing, more specific routes to IP address ranges that point to an internet gateway or a NAT device, those routes take precedence. If you have existing routes destined for an IP address range that is identical to the IP address range used by the service, then your routes take precedence.

Example: An endpoint route in a route table

In this scenario, you have an existing route in your route table for all internet traffic (`0.0.0.0/0`) that points to an internet gateway. Any traffic from the subnet that's destined for another AWS service uses the internet gateway.

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-1a2b3c4d

You create an endpoint to a supported AWS service, and associate your route table with the endpoint. An endpoint route is automatically added to the route table, with a destination of `pl-1a2b3c4d` (assume this represents the service to which you've created the endpoint). Now, any traffic from the subnet that's destined for that AWS service in the same Region goes to the endpoint, and does not go to the internet gateway. All other internet traffic goes to your internet gateway, including traffic that's destined for other services, and destined for the AWS service in other Regions.

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-1a2b3c4d
pl-1a2b3c4d	vpce-11bb22cc

Example: Adjusting your route tables for endpoints

In this scenario, `54.123.165.0/24` is in the Amazon S3 IP address range and you configured your route table to allow instances in your subnet to communicate with Amazon S3 buckets through an internet gateway. You've added a route with `54.123.165.0/24` as a destination, and the internet gateway as the target. You then create an endpoint, and associate this route table with the endpoint. An endpoint route is automatically added to the route table. You then use the describe-prefix-lists command to view the IP address range for Amazon S3. The range is `54.123.160.0/19`, which is less specific than the range that's pointing to your internet gateway. This means that any traffic destined for the `54.123.165.0/24` IP address range continues to use the internet gateway, and does not use the endpoint (for as long as this remains the public IP address range for Amazon S3).

Destination	Target
10.0.0.0/16	Local
54.123.165.0/24	igw-1a2b3c4d
pl-1a2b3c4d	vpce-11bb22cc

To ensure that all traffic destined for Amazon S3 in the same Region is routed via the endpoint, you must adjust the routes in your route table. To do this, you can delete the route to the internet gateway. Now, all traffic to Amazon S3 in the same Region uses the endpoint, and the subnet that's associated with your route table is a private subnet.

Destination	Target
10.0.0.0/16	Local
pl-1a2b3c4d	vpce-11bb22cc

Gateway endpoint limitations

To use gateway endpoints, you need to be aware of the current limitations:

- You cannot use an AWS prefix list ID in an outbound rule in a network ACL to allow or deny outbound traffic to the service specified in an endpoint. If your network ACL rules restrict traffic, you must specify the CIDR block (IP address range) for the service instead. You can, however, use an AWS prefix list ID in an outbound security group rule. For more information, see Security groups (p. 38).

- Endpoints are supported within the same Region only. You cannot create an endpoint between a VPC and a service in a different Region.
- Endpoints support IPv4 traffic only.
- You cannot transfer an endpoint from one VPC to another, or from one service to another.
- You have a quota on the number of endpoints you can create per VPC. For more information, see AWS PrivateLink quotas (p. 72).
- Endpoint connections cannot be extended out of a VPC. Resources on the other side of a VPN connection, VPC peering connection, transit gateway, AWS Direct Connect connection, or ClassicLink connection in your VPC cannot use the endpoint to communicate with resources in the endpoint service.
- You must turn on DNS resolution in your VPC, or if you're using your own DNS server, ensure that DNS requests to the required service (such as Amazon S3) are resolved correctly to the IP addresses maintained by AWS. For more information, see Using DNS with your VPC in the *Amazon VPC User Guide* and AWS IP Address Ranges in the *Amazon Web Services General Reference*.
- Review the service-specific limits for your endpoint service.

For more information about rules and limitations that are specific to Amazon S3, see Endpoints for Amazon S3 (p. 25).

For more information about rules and limitations that are specific to DynamoDB, see Endpoints for Amazon DynamoDB (p. 31).

Endpoints for Amazon S3

If you've already set up access to your Amazon S3 resources from your VPC, you can continue to use Amazon S3 DNS names to access those resources after you've set up an endpoint. However, take note of the following:

- Your endpoint has a policy that controls the use of the endpoint to access Amazon S3 resources. The default policy allows access by any user or service within the VPC, using credentials from any AWS account, to any Amazon S3 resource; including Amazon S3 resources for an AWS account other than the account with which the VPC is associated. For more information, see Control access to services with VPC endpoints (p. 37).
- The source IPv4 addresses from instances in your affected subnets as received by Amazon S3 change from public IPv4 addresses to the private IPv4 addresses in your VPC. An endpoint switches network routes, and disconnects open TCP connections. The previous connections that used public IPv4 addresses are not resumed. We recommend that you do not have any critical tasks running when you create or modify an endpoint; or that you test to ensure that your software can automatically reconnect to Amazon S3 after the connection break.
- You cannot use an IAM policy or bucket policy to allow access from a VPC IPv4 CIDR range (the private IPv4 address range). VPC CIDR blocks can be overlapping or identical, which may lead to unexpected results. Therefore, you cannot use the `aws:SourceIp` condition in your IAM policies for requests to Amazon S3 through a VPC endpoint. This applies to IAM policies for users and roles, and any bucket policies. If a statement includes the `aws:SourceIp` condition, the value fails to match any provided IP address or range. Instead, you can do the following:
  - Use your route tables to control which instances can access resources in Amazon S3 via the endpoint.
  - For bucket policies, you can restrict access to a specific endpoint or to a specific VPC. For more information, see Amazon S3 bucket policies (p. 29).
- Endpoints currently do not support cross-Region requests—ensure that you create your endpoint in the same Region as your bucket. You can find the location of your bucket by using the Amazon S3 console, or by using the get-bucket-location command. Use a Region-specific Amazon S3 endpoint to access your bucket; for example, `mybucket.s3.us-west-2.amazonaws.com`. For more information about Region-specific endpoints for Amazon S3, see Amazon Simple Storage Service (S3) in *Amazon*

Web Services General Reference. If you use the AWS CLI to make requests to Amazon S3, set your default Region to the same Region as your bucket, or use the `--region` parameter in your requests.

Note
Treat the US Standard Region for Amazon S3 as mapped to the `us-east-1` Region.

- Endpoints are currently supported for IPv4 traffic only.

Before you use endpoints with Amazon S3, ensure that you have also read the following general limitations: Gateway endpoint limitations (p. 24). For information about creating and viewing S3 buckets, see How Do I Create an S3 Bucket and How Do I View the Properties for an S3 Bucket in the *Amazon Simple Storage Service Console User Guide*.

If you use other AWS services in your VPC, they might use S3 buckets for certain tasks. Ensure that your endpoint policy allows full access to Amazon S3 (the default policy), or that it allows access to the specific buckets that are used by these services. Alternatively, only create an endpoint in a subnet that is not used by any of these services, to allow the services to continue accessing S3 buckets using public IP addresses.

The following table lists AWS services that might be affected by an endpoint, and any specific information for each service.

AWS service	Note
Amazon AppStream 2.0	Your endpoint policy must allow access to the specific buckets that are used by AppStream 2.0 for storing user content. For more information, see Using Amazon S3 VPC Endpoints for Home Folders and Application Settings Persistence in the *Amazon AppStream 2.0 Administration Guide*.
AWS CloudFormation	If you have resources in your VPC that must respond to a wait condition or custom resource request, your endpoint policy must allow at least access to the specific buckets that are used by these resources. For more information, see Setting Up VPC Endpoints for AWS CloudFormation.
CodeDeploy	Your endpoint policy must allow full access to Amazon S3, or allow access to any S3 buckets that you've created for your CodeDeploy deployments.
Elastic Beanstalk	Your endpoint policy must allow at least access to any S3 buckets used for Elastic Beanstalk applications. For more information, see Using Elastic Beanstalk with Amazon S3 in the *AWS Elastic Beanstalk Developer Guide*.
Amazon EMR	Your endpoint policy must allow access to the Amazon Linux repositories and other buckets that are used by Amazon EMR. For more information, see Minimum Amazon S3 Policy for Private Subnet in the *Amazon EMR Management Guide*.
AWS OpsWorks	Your endpoint policy must allow at least access to specific buckets that are used by AWS OpsWorks. For more information, see Running a Stack in a VPC in the *AWS OpsWorks User Guide*.

AWS service	Note
AWS Systems Manager	Your endpoint policy must allow access to the Amazon S3 buckets used by Patch Manager for patch baseline operations in your AWS Region. These buckets contain the code that is retrieved and run on instances by the patch baseline service. For more information, see Create a Virtual Private Cloud Endpoint in the *AWS Systems Manager User Guide*. For a list of S3 bucket permissions required by SSM Agent for its operations, see Minimum S3 Bucket Permissions for SSM Agent in the *AWS Systems Manager User Guide*.
Amazon Elastic Container Registry	Your endpoint policy must allow access to the Amazon S3 buckets used by Amazon ECR to store Docker image layers. For more information, see Minimum Amazon S3 Bucket Permissions for Amazon ECR in the *Amazon Elastic Container Registry User Guide*.
Amazon WorkDocs	If you use an Amazon WorkDocs client in Amazon WorkSpaces or an EC2 instance, your endpoint policy must allow full access to Amazon S3.
Amazon WorkSpaces	Amazon WorkSpaces does not directly depend on Amazon S3. However, if you provide Amazon WorkSpaces users with internet access, then take note that websites, HTML emails, and internet services from other companies may depend on Amazon S3. Ensure that your endpoint policy allows full access to Amazon S3 to allow these services to continue to work correctly.

Traffic between your VPC and S3 buckets does not leave the Amazon network.

Endpoint policies for Amazon S3

The following are example endpoint policies for accessing Amazon S3. For more information, see Use VPC endpoint policies (p. 37). It is up to the user to determine the policy restrictions that meet their business needs.

> **Important**
> All types of policies — IAM user policies, endpoint policies, S3 bucket policies, and Amazon S3 ACL policies (if any) — must grant the necessary permissions for access to Amazon S3 to succeed.
> AWS recommends that you use IAM conditions, rather than the IAM `Principal` element, in VPC endpoint policies when you are restricting use of the endpoint to particular callers. Examples of such conditions are `aws:PrincipalArn`, `aws:PrincipalAccount`, `aws:PrincipalOrgId`, and `aws:PrincipalOrgPaths`. For more information about condition context keys, see AWS global condition context keys in the *AWS Identity and Access Management User Guide*.

Example Example: Restricting 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. The following is an example of a policy that restricts access to `my_secure_bucket` only.

```
{
  "Effect": "Allow",
  "Action": ["s3:ListBucket","s3:GetObject","s3:PutObject"],
  "Resource": ["arn:aws:s3:::example-bucket","arn:aws:s3:::example-bucket/*"]
}
```

Example Example: Restricting use of this VPC endpoint to a specific IAM role in an account

You can create a policy that restricts use of the VPC endpoint to a specific IAM role. The following is an example that restricts the access to `SomeRole` in account `111122223333`.

```
{
  "Sid": "Restrict-acess-to-specific-IAM-role",
  "Effect": "Allow",
  "Principal": "*",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "ArnEquals": {
      "aws:PrincipalArn": "arn:aws:iam::111122223333:role/SomeRole"
    }
  }
}
```

Example Example: Restricting use of this VPC endpoint to a users in a specific account

You can create a policy that restricts use of the VPC endpoint to a specific account. The following is an example that restricts the access to users in account `111122223333`.

```
{
  "Sid": "AllowCallersFromAccount111122223333",
  "Effect": "Allow",
  "Principal": "*",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:PrincipalAccount": "111122223333"
    }
  }
}
```

Example Example: Enabling access to the Amazon Linux AMI repositories

The Amazon Linux AMI repositories are Amazon S3 buckets in each Region. If you want instances in your VPC to access the repositories through an endpoint, create an endpoint policy that enables access to these buckets.

The following policy allows access to the Amazon Linux repositories.

You need to replace `region` with your AWS Region, for example, **us-east-1**.

```
{
  "Statement": [
```

```
    {
      "Sid": "AmazonLinuxAMIRepositoryAccess",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::packages.region.amazonaws.com/*",
        "arn:aws:s3:::repo.region.amazonaws.com/*"
      ]
    }
  ]
}
```

The following policy allows access to the Amazon Linux 2 repositories.

You need to replace `region` with your AWS Region, for example, **us-east-1**.

```
{
  "Statement": [
    {
      "Sid": "AmazonLinux2AMIRepositoryAccess",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::amazonlinux.region.amazonaws.com/*"
      ]
    }
  ]
}
```

Amazon S3 bucket policies

You can use bucket policies to control access to buckets from specific endpoints, or specific VPCs.

You cannot use the `aws:SourceIp` condition in your bucket policies for requests to Amazon S3 through a VPC endpoint. The condition fails to match any specified IP address or IP address range, and may have an undesired effect when you make requests to an Amazon S3 bucket. For example:

- You have a bucket policy with a `Deny` effect and a `NotIpAddress` condition that's intended to grant access from a single or limited range of IP addresses only. For requests to the bucket through an endpoint, the `NotIpAddress` condition is always matched, and the statement's effect applies, assuming other constraints in the policy match. Access to the bucket is denied.
- You have a bucket policy with a `Deny` effect and an `IpAddress` condition that's intended to deny access to a single or limited range of IP addresses only. For requests to the bucket through an endpoint, the condition is not matched, and the statement does not apply. Access to the bucket is allowed, assuming there are other statements that allow access without an `IpAddress` condition.

Adjust your bucket policy to limit access to a specific VPC or a specific VPC endpoint instead.

For more information about bucket policies for Amazon S3, see Using Bucket Policies and User Policies in *Amazon Simple Storage Service Developer Guide*.

The following are example bucket policies that limit access to a specific VPC endpoint or specific VPC. To enable IAM users to work with bucket policies, you must grant them permission to use the `s3:GetBucketPolicy` and `s3:PutBucketPolicy` actions.

Example Example: Restricting access to a specific endpoint

The following is an example of an S3 bucket policy that allows access to a specific bucket, `my_secure_bucket`, from endpoint `vpce-1a2b3c4d` only. The policy denies all access to the bucket if the specified endpoint is not being used. The `aws:sourceVpce` condition is used to specify the endpoint. The `aws:sourceVpce` condition does not require an ARN for the VPC endpoint resource, only the endpoint ID.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    {
      "Sid": "Access-to-specific-VPCE-only",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::my_secure_bucket",
                   "arn:aws:s3:::my_secure_bucket/*"],
      "Condition": {
        "StringNotEquals": {
          "aws:sourceVpce": "vpce-1a2b3c4d"
        }
      }
    }
  ]
}
```

Example Example: Restricting access to a specific VPC

You can create a bucket policy that restricts access to a specific VPC by using the `aws:sourceVpc` condition. This is useful if you have multiple endpoints configured in the same VPC, and you want to manage access to your S3 buckets for all of your endpoints. The following is an example of a policy that allows VPC `vpc-111bbb22` to access `my_secure_bucket` and its objects. The policy denies all access to the bucket if the specified VPC is not being used. The `aws:sourceVpc` condition does not require an ARN for the VPC resource, only the VPC ID.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    {
      "Sid": "Access-to-specific-VPC-only",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::my_secure_bucket",
                   "arn:aws:s3:::my_secure_bucket/*"],
      "Condition": {
        "StringNotEquals": {
          "aws:sourceVpc": "vpc-111bbb22"
        }
      }
    }
  ]
}
```

Example Example: Restricting access to buckets in a specific AWS Account

You can create a policy that restricts access only to the S3 buckets in a specific AWS Account. This is useful if you would like to restrict clients within your VPC from accessing buckets that you do not own.

The following is an example of a policy that restricts access to resources owned by a single AWS Account, with the account ID of `111122223333`.

```
{
  "Statement": [
    {
      "Sid": "Access-to-bucket-in-specific-account-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "StringNotEquals": {
          "s3:ResourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

Endpoints for Amazon DynamoDB

If you've already set up access to your DynamoDB tables from your VPC, you can continue to access the tables as you normally would after you set up a gateway endpoint. However, take note of the following:

- Your endpoint has a policy that controls the use of the endpoint to access DynamoDB resources. The default policy allows access by any user or service within the VPC, using credentials from any AWS account, to any DynamoDB resource. For more information, see Control access to services with VPC endpoints (p. 37).
- DynamoDB does not support resource-based policies (for example, on tables). Access to DynamoDB is controlled through the endpoint policy and IAM policies for individual IAM users and roles.
- Endpoints currently do not support cross-region requests—ensure that you create your endpoint in the same Region as your DynamoDB tables.
- If you use AWS CloudTrail to log DynamoDB operations, the log files contain the private IP address of the EC2 instance in the VPC and the endpoint ID for any actions performed through the endpoint.
- The source IPv4 addresses from instances in your affected subnets change from public IPv4 addresses to the 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 an endpoint; or that you test to ensure that your software can automatically reconnect to DynamoDB after the connection break.

Before you use endpoints with DynamoDB, ensure that you have also read the following general limitations: Gateway endpoint limitations (p. 24).

For more information about creating a gateway VPC endpoint, see Gateway VPC endpoints (p. 21).

Endpoint policies for DynamoDB

An endpoint policy is an IAM policy that you attach to an endpoint that allows access to some or all of the service to which you're connecting. The following are example endpoint policies for accessing DynamoDB.

Important
All types of policies — IAM user policies and endpoint policies — must grant the necessary permissions for access to DynamoDB to succeed.

Example Example: Read-only access

You can create a policy that restricts actions to only listing and describing DynamoDB tables through the VPC endpoint.

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

Example Example: Restrict access to a specific table

You can create a policy that restricts access to a specific DynamoDB table. In this example, the endpoint policy allows access to `StockTable` only.

```
{
  "Statement": [
    {
      "Sid": "AccessToSpecificTable",
      "Principal": "*",
      "Action": [
        "dynamodb:Batch*",
        "dynamodb:Delete*",
        "dynamodb:DescribeTable",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:Update*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:dynamodb:us-east-1:123456789012:table/StockTable"
    }
  ]
}
```

Use IAM policies to control access to DynamoDB

You can create an IAM policy for your IAM users, groups, or roles to restrict access to DynamoDB tables from a specific VPC endpoint only. To do this, you can use the `aws:sourceVpce` condition key for the table resource in your IAM policy.

For more information about managing access to DynamoDB, see Authentication and Access Control for Amazon DynamoDB in the *Amazon DynamoDB Developer Guide*.

Example Example: Restrict access from a specific endpoint

In this example, users are denied permission to work with DynamoDB tables, except if accessed through endpoint `vpce-11aa22bb`.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AccessFromSpecificEndpoint",
         "Action": "dynamodb:*",
         "Effect": "Deny",
         "Resource": "arn:aws:dynamodb:region:account-id:table/*",
         "Condition": { "StringNotEquals" : { "aws:sourceVpce": "vpce-11aa22bb" } }
      }
   ]
}
```

Example Example: Restricting use of this VPC endpoint to a specific IAM role in an account

You can create a policy that restricts use of the VPC endpoint to a specific IAM role. The following is an example that restricts the access to `SomeRole` in account `111122223333`.

```
{
  "Sid": "Restrict-acess-to-specific-IAM-role",
  "Effect": "Allow",
  "Principal": "*",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "ArnEquals": {
      "aws:PrincipalArn": "arn:aws:iam::111122223333:role/SomeRole"
    }
  }
}
```

Example Example: Restricting use of this VPC endpoint to a users in a specific account

You can create a policy that restricts use of the VPC endpoint to a specific account. The following is an example that restricts the access to users in account `111122223333`.

```
{
  "Sid": "AllowCallersFromAccount111122223333",
  "Effect": "Allow",
  "Principal": "*",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:PrincipalAccount": "111122223333"
    }
  }
}
```

Create a gateway endpoint

To create an endpoint, you must specify the VPC in which you want to create the endpoint, and the service to which you want to establish the connection.

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**, **Create Endpoint**.

3. For **Service Name**, choose the service to which to connect. To create a gateway endpoint to DynamoDB or Amazon S3, ensure that the **Type** column indicates **Gateway**.
4. Complete the following information, and choose **Create endpoint**.
   - For **VPC**, select a VPC in which to create the endpoint.
   - For **Configure 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 to the selected route tables.
   - For **Policy**, choose the type of policy. You can leave the default option, **Full Access**, to allow full access to the service. Alternatively, you can select **Custom**, and then use the AWS Policy Generator to create a custom policy, or enter your own policy in the policy window.
   - (Optional) Add or remove a tag.

     [Add a tag] Choose **Add tag** and do the following:
     - For **Key**, enter the key name.
     - For **Value**, enter the key value.

     [Remove a tag] Choose the delete button ("x") to the right of the tag's Key and Value.

After you've created an endpoint, you can view information about it.

To view information about 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** and select your endpoint.
3. To view information about the endpoint, choose **Summary**. You can get the AWS prefix list name for the service in the **Service** box.
4. To view information about the route tables that are used by the endpoint, choose **Route Tables**.
5. To view the IAM policy that's attached to your endpoint, choose **Policy**.

   Note
   The **Policy** tab only displays the endpoint policy. It does not display any information about IAM policies for IAM users that have permission to work with endpoints. It also does not display service-specific policies; for example, S3 bucket policies.

To create and view an endpoint using the AWS CLI

1. Use the describe-vpc-endpoint-services command to get a list of available services. In the output that's returned, take note of the name of the service to which you want to connect. The `serviceType` field indicates whether you connect to the service via an interface endpoint or a gateway endpoint.

```
aws ec2 describe-vpc-endpoint-services
```

```
{
    "serviceDetailSet": [
        {
            "serviceType": [
                {
                    "serviceType": "Gateway"
                }
            ...
```

2. To create a gateway endpoint (for example, to Amazon S3), use the create-vpc-endpoint command and specify the VPC ID, service name, and route tables that will use the endpoint. You can optionally

use the `--policy-document` parameter to specify a custom policy to control access to the service. If the parameter is not used, we attach a default policy that allows full access to the service.

For Amazon S3, you must set the `--vpc-endpoint-type` parameter to `Gateway`.

```
aws ec2 create-vpc-endpoint --vpc-id vpc-1a2b3c4d --service-name com.amazonaws.us-east-1.s3 --route-table-ids rtb-11aa22bb --vpc-endpoint-type Gateway
```

3. Describe your endpoint using the describe-vpc-endpoints command.

```
aws ec2 describe-vpc-endpoints
```

To describe available services using the AWS Tools for Windows PowerShell or API

- Get-EC2VpcEndpointService (AWS Tools for Windows PowerShell)
- DescribeVpcEndpointServices (Amazon EC2 Query API)

To create a VPC endpoint using the AWS Tools for Windows PowerShell or API

- New-EC2VpcEndpoint (AWS Tools for Windows PowerShell)
- CreateVpcEndpoint (Amazon EC2 Query API)

To describe your VPC endpoints using the AWS Tools for Windows PowerShell or API

- Get-EC2VpcEndpoint (AWS Tools for Windows PowerShell)
- DescribeVpcEndpoints (Amazon EC2 Query API)

Modify your security group

If the VPC security group associated with your instance restricts outbound traffic, you must add a rule to allow traffic destined for the AWS service to leave your instance.

To add an outbound rule for a gateway endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Security Groups**.
3. Select your VPC security group, choose the **Outbound rules** tab, and then choose **Edit outbound rules**.
4. Select the type of traffic from the **Type** list, and enter the port range, if required. For example, if you use your instance to retrieve objects from Amazon S3, choose **HTTPS** from the **Type** list.
5. For **Destination**, start entering `pl-` to display a list of prefix list IDs and names for the available AWS services. Choose the prefix list ID for the AWS service, or enter it.
6. Choose **Save**.

To get the prefix list name, ID, and IP address range for an AWS service using the command line or API

- describe-prefix-lists (AWS CLI)
- Get-EC2PrefixList (AWS Tools for Windows PowerShell)

- DescribePrefixLists (Amazon EC2 Query API)

Modify a gateway endpoint

You can modify a gateway endpoint by changing or removing its policy, and adding or removing the route tables that are used by the endpoint.

If you want to migrate an existing Amazon S3 gateway endpoint to an interface endpoint, after you create the Amazon S3 interface endpoint, delete the Amazon S3 gateway endpoint. For more information, see the section called "Create an interface endpoint" (p. 10) and the section called "Delete a VPC endpoint" (p. 38).

To change the policy associated with a gateway 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**, **Edit policy**.
4. You can choose **Full Access** to allow full access. Alternatively, choose **Custom**, and then use the AWS Policy Generator to create a custom policy, or enter your own policy in the policy window. When you're done, choose **Save**.

   Note
   It can take a few minutes for policy changes to take effect.

To add or remove route tables used by a gateway 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**, **Manage Route Tables**.
4. Select or deselect the required route tables, and choose **Modify Route Tables**.

To modify a gateway endpoint using the AWS CLI

1. Use the describe-vpc-endpoints command to get the ID of your gateway endpoint.

```
aws ec2 describe-vpc-endpoints
```

2. The following example uses the modify-vpc-endpoint command to associate route table `rtb-aaa222bb` with the gateway endpoint, and reset the policy document.

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-1a2b3c4d --add-route-table-ids rtb-aaa222bb --reset-policy
```

To modify a VPC endpoint using the AWS Tools for Windows PowerShell or an API

- Edit-EC2VpcEndpoint (AWS Tools for Windows PowerShell)
- ModifyVpcEndpoint (Amazon EC2 Query API)

Add or remove gateway endpoint tags

Tags provide a way to identify the gateway endpoint. You can add or remove a tag.

To add or remove a gateway endpoint tag

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 and choose **Actions**, **Add/Edit Tags**.
4. Add or remove a tag.

   [Add a tag] Choose **Create tag** and do the following:
   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Choose the delete button ("x") to the right of the tag's Key and Value.

To add or remove a tag using the AWS Tools for Windows PowerShell or an API

- create-tags (AWS CLI)
- CreateTags (AWS Tools for Windows PowerShell)
- delete-tags (AWS CLI)
- DeleteTags (AWS Tools for Windows PowerShell)

Control access to services with VPC endpoints

When you create an interface or gateway endpoint, you can attach an endpoint policy to it that controls access to the service to which you are connecting. Endpoint policies must be written in JSON format. Not all services support endpoint policies.

If you're using an endpoint to Amazon S3, you can also use Amazon S3 bucket policies to control access to buckets from specific endpoints, or specific VPCs. For more information, see Amazon S3 bucket policies (p. 29).

Contents

Use VPC endpoint policies

A VPC endpoint policy is an IAM resource policy that you attach to an endpoint when you create or modify the endpoint. If you do not attach a policy when you create an endpoint, we attach a default policy for you that allows full access to the service. If a service does not support endpoint policies, the endpoint allows full access to the service. An endpoint policy does not override or replace IAM user policies or service-specific policies (such as S3 bucket policies). It is a separate policy for controlling access from the endpoint to the specified service.

You cannot attach more than one policy to an endpoint. However, you can modify the policy at any time. If you do modify a policy, it can take a few minutes for the changes to take effect. For more information about writing policies, see Overview of IAM Policies in the *IAM User Guide*.

Your endpoint policy can be like any IAM policy; however, take note of the following:

- Your policy must contain a Principal element. For additional information related gateway endpoints, see Endpoint policies for gateway endpoints (p. 38).
- The size of an endpoint policy cannot exceed 20,480 characters (including white space).

For information about the services that support endpoint policies, see *Services that support AWS PrivateLink* (p. 67).

Endpoint policies for gateway endpoints

For endpoint polices that are applied to gateway endpoints, if you specify `Principal` in the format `"AWS":"AWS-account-ID"` or `"AWS":"arn:aws:iam::AWS-account-ID:root"`, access is granted to the account root user only, and not all IAM users and roles for the account.

If you specify an Amazon Resource Name (ARN) for the `Principal` element, the ARN is transformed to a unique principal ID when the policy is saved.

For example endpoint policies for Amazon S3 and DynamoDB, see the following topics:

- Endpoint policies for Amazon S3 (p. 27)
- Endpoint policies for DynamoDB (p. 31)

Security groups

When you create an interface endpoint, you can associate security groups with the endpoint network interface that is created in your VPC. If you do not specify a security group, the default security group for your VPC is automatically associated with the endpoint network interface. You must ensure that the rules for the security group allow communication between the endpoint network interface and the resources in your VPC that communicate with the service.

For a gateway endpoint, if your security group's outbound rules are restricted, you must add a rule that allows outbound traffic from your VPC to the service that's specified in your endpoint. To do this, you can use the service's AWS prefix list ID as the destination in the outbound rule. For more information, see Modify your security group (p. 35).

Security groups do not apply to Gateway Load Balancer endpoints.

Delete a VPC endpoint

If you no longer require an endpoint, you can delete it. Deleting a gateway endpoint also deletes the endpoint routes in the route tables that were used by the endpoint, but doesn't affect any security groups associated with the VPC in which the endpoint resides. Deleting an interface endpoint or Gateway Load Balancer endpoint also deletes the endpoint network interfaces.

A Gateway Load Balancer endpoint cannot be deleted if there are routes in your route tables that point to the endpoint.

To delete an 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 VPC endpoint

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

- DeleteVpcEndpoints (Amazon EC2 Query API)

VPC endpoint services (AWS PrivateLink)

You can create your own application in your VPC and configure it as an AWS PrivateLink-powered service (referred to as an *endpoint service*). Other AWS principals can create a connection from their VPC to your endpoint service using an interface VPC endpoint (p. 3) or a Gateway Load Balancer endpoint (p. 18), depending on the type of service. You are the *service provider*, and the AWS principals that create connections to your service are *service consumers*.

Contents

VPC endpoint services for interface endpoints

The following are the general steps to create an endpoint service for interface endpoints.

1. Create a Network Load Balancer for your application in your VPC and configure it for each subnet (Availability Zone) in which the service should be available. The load balancer receives requests from service consumers and routes it to your service. For more information, see Getting started with Network Load Balancers in the *User Guide for Network Load Balancers*.

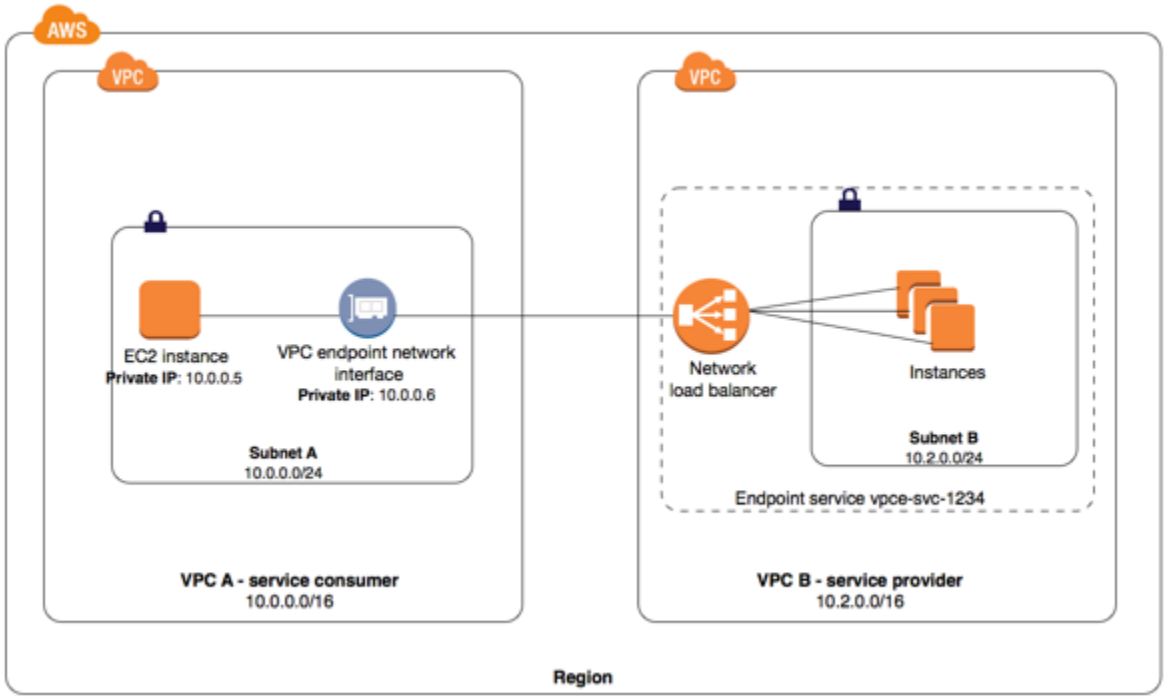
   We recommend that you configure your service in all Availability Zones within the Region.
2. Create a VPC endpoint service configuration and specify your Network Load Balancer.

The following are the general steps to enable service consumers to connect to your service.

1. Grant permissions to specific service consumers (AWS accounts, IAM users, and IAM roles) to create a connection to your endpoint service.
2. A service consumer that has been granted permissions creates an interface endpoint to your service, optionally in each Availability Zone in which you configured your service.
3. To activate the connection, accept the interface endpoint connection request. By default, connection requests must be manually accepted. However, you can configure the acceptance settings for your endpoint service so that any connection requests are automatically accepted.

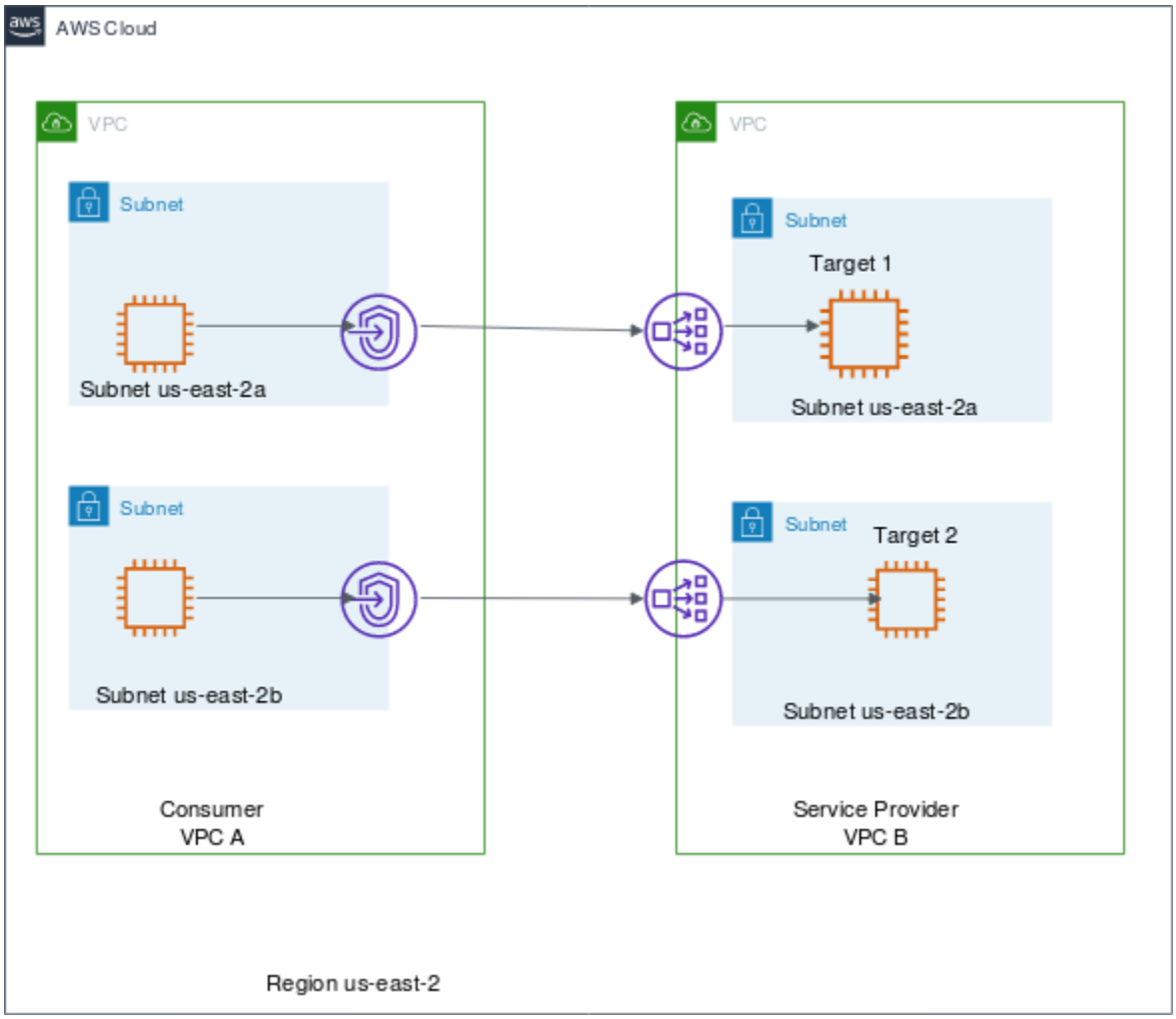
The combination of permissions and acceptance settings can help you control which service consumers (AWS principals) can access your service. For example, you can grant permissions to selected principals that you trust and automatically accept all connection requests, or you can grant permissions to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

In the following diagram, the account owner of VPC B is a service provider, and has a service running on instances in subnet B. The owner of VPC B has a service endpoint (vpce-svc-1234) with an associated Network Load Balancer that points to the instances in subnet B as targets. Instances in subnet A of VPC A use an interface endpoint to access the services in subnet B.



For low latency and fault tolerance, we recommend using a Network Load Balancer with targets in every Availability Zone of the AWS Region. To help achieve high availability for service consumers that use zonal DNS hostnames (p. 15) to access the service, you can enable cross-zone load balancing. Cross-zone load balancing enables the load balancer to distribute traffic across the registered targets in all enabled Availability Zones. For more information, see Cross-Zone Load Balancing in the *User Guide for Network Load Balancers*. Regional data transfer charges may apply to your account when you enable cross-zone load balancing.

In the following diagram, the owner of VPC B is the service provider, and it has configured a Network Load Balancer with targets in two different Availability Zones. The service consumer (VPC A) has created interface endpoints in the same two Availability Zones in their VPC. Requests to the service from instances in VPC A can use either interface endpoint.



For examples of configuring a service and enabling service consumers to access it over a VPC peering connection, see Examples: Services using AWS PrivateLink and VPC peering in the *Amazon VPC User Guide*.

Endpoint service Availability Zone considerations

When you create an endpoint service, the service is created in the Availability Zone that is mapped to your account and is independent from other accounts. When the service provider and the consumer are in different accounts, use the Availability Zone ID to uniquely and consistently identify the endpoint service Availability Zone. For example, `use1-az1` is an AZ ID for the `us-east-1` Region and maps to the same location in every AWS account. For information about Availability Zone IDs, see AZ IDs for Your Resources in the *AWS RAM User Guide* or use describe-availability-zones.

When the service provider and the consumer have different accounts and use multiple Availability Zones, and the consumer views the VPC endpoint service information, the response only includes the common Availability Zones. For example, when the service provider account uses `us-east-1a` and `us-east-1c` and the consumer uses `us-east-1a` and `us-east-1b`, the response includes the VPC endpoint services in the common Availability Zone, `us-east-1a`.

Endpoint service DNS names

When you create a VPC endpoint service, AWS generates endpoint-specific DNS hostnames that you can use to communicate with the service. These names include the VPC endpoint ID, the Availability Zone name and Region Name, for example, vpce-1234-abcdev-us-east-1.vpce-svc-123345.us-east-1.vpce.amazonaws.com. By default, your consumers access the service with that DNS name and usually need to modify the application configuration.

If the endpoint service is for an AWS service, or a service available in the AWS Marketplace, there is a default DNS name. For other services, the service provider can configure a private DNS name so consumers can access the service using an existing DNS name without making changes to their applications. For more information, see *Private DNS names* (p. 58).

Service providers can use the **ec2:VpceServicePrivateDnsName** condition context key in an IAM policy statement to control what private DNS names can be created. For more information, see Actions defined by Amazon EC2 in the *IAM User Guide*.

Private DNS name requirements

Service providers can specify a private DNS name for a new endpoint service, or an existing endpoint service. To use a private DNS name, enable the feature, and then specify a private DNS name. Before consumers can use the private DNS name, you must verify that you have control of the domain/subdomain. You can initiate domain ownership verification using the Amazon VPC Console or API. After the domain ownership verification completes, consumers access the endpoint by using the private DNS name.

Connect to on-premises data centers

You can use the following types of connections for a connection between an interface endpoint and your on-premises data center:

- AWS Direct Connect
- AWS Site-to-Site VPN

Access services through a VPC peering connection

You can use a VPC peering connection with a VPC endpoint to allow private access to consumers across the VPC peering connection. For more information, see Examples: Services using AWS PrivateLink and VPC peering in the *Amazon VPC User Guide*.

Use proxy protocol for connection information

A Network Load Balancer provides source IP addresses to your application (your service). When service consumers send traffic to your service through an interface endpoint, the source IP addresses provided to your application are the private IP addresses of the Network Load Balancer nodes, and not the IP addresses of the service consumers.

If you need the IP addresses of the service consumers and their corresponding interface endpoint IDs, enable Proxy Protocol on your load balancer and get the client IP addresses from the Proxy Protocol header. For more information, see Proxy protocol in the *User Guide for Network Load Balancers*.

Rules and limitations

To use endpoint services, you need to be aware of the current rules and limitations:

- An endpoint service supports IPv4 traffic over TCP only.
- Service consumers can use the endpoint-specific DNS hostnames to access the endpoint service, or the private DNS name.
- If an endpoint service is associated with multiple Network Load Balancers, then for a specific Availability Zone, an interface endpoint establishes a connection with one load balancer only.
- For the endpoint service, the associated Network Load Balancer can support 55,000 simultaneous connections or about 55,000 connections per minute to each unique target (IP address and port). If you exceed these connections, there is an increased chance of port allocation errors. To fix the port allocation errors, add more targets to the target group. For information about Network Load Balancer target groups, see Target groups for your Network Load Balancers and Register targets with your Target Group in the *User Guide for Network Load Balancers*.
- Availability Zones in your account might not map to the same locations as Availability Zones in another account. For example, your Availability Zone `us-east-1a` might not be the same location as `us-east-1a` for another account. For more information, see Regions and Zones. When you configure an endpoint service, it's configured in the Availability Zones as mapped to your account.
- An endpoint service is only available in the Region where you created it.
- Review the service-specific limits for your endpoint service.
- Review the security best practices and examples for endpoint services. For more information, see Policy best practices and the section called “Control access to services” (p. 37).

VPC endpoint services for Gateway Load Balancer endpoints

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 can then configure the Gateway Load Balancer as a VPC endpoint service, to enable other AWS principals to access the service through a Gateway Load Balancer endpoint.

The following are the general steps to create an endpoint service for a Gateway Load Balancer endpoint.

1. Create a Gateway Load Balancer for your virtual appliances. For more information, see Getting started with Gateway Load Balancers.

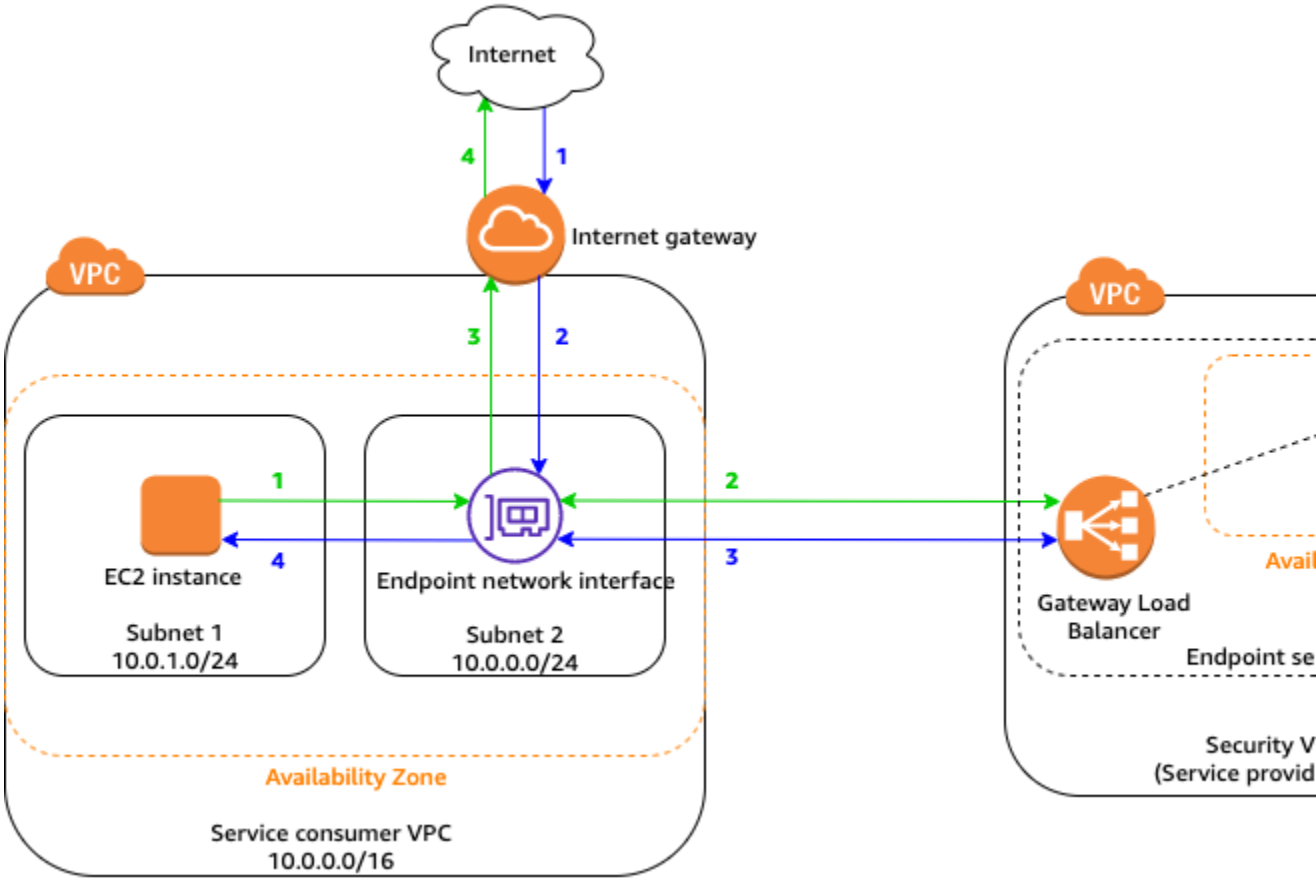
   We recommend that you configure your service in all Availability Zones within the Region.
2. Create a VPC endpoint service configuration and specify your Gateway Load Balancer.

The following are the general steps to enable service consumers to connect to your service.

1. Grant permissions to specific service consumers (AWS accounts, IAM users, and IAM roles) to create a connection to your endpoint service.
2. A service consumer that has been granted permissions creates a Gateway Load Balancer endpoint (p. 18) to your service.
3. To activate the connection, accept the endpoint connection request. By default, connection requests must be manually accepted. However, you can configure the acceptance settings for your endpoint service so that any connection requests are automatically accepted.

In the following example, a fleet of security appliances is configured behind a Gateway Load Balancer in the security VPC. An endpoint service is configured for the Gateway Load Balancer. The owner of the service consumer VPC creates a Gateway Load Balancer endpoint in subnet 2 in their VPC (represented by an endpoint network interface). All traffic entering the VPC through the internet gateway is first

routed to the Gateway Load Balancer endpoint for inspection in the security VPC before it's routed to the destination subnet. Similarly, all traffic leaving the EC2 instance in subnet 1 is first routed to Gateway Load Balancer endpoint for inspection in the security VPC before it's routed to the internet.



For more information about the routing configuration for this scenario, see Routing to a Gateway Load Balancer endpoint in the *Amazon VPC User Guide*.

Availability Zone considerations

When you create an endpoint service, the service is created in the Availability Zone that is mapped to your account and is independent from other accounts. When the service provider and the consumer are in different accounts, use the Availability Zone ID to uniquely and consistently identify the endpoint service Availability Zone. For example, `use1-az1` is an AZ ID for the `us-east-1` Region and maps to the same location in every AWS account. For information about Availability Zone IDs, see AZ IDs for Your Resources in the *AWS RAM User Guide* or use describe-availability-zones.

When the service provider and the consumer have different accounts and use multiple Availability Zones, and the consumer views the VPC endpoint service information, the response only includes the common Availability Zones. For example, when the service provider account uses `us-east-1a` and `us-east-1c` and the consumer uses `us-east-1a` and `us-east-1b`, the response includes the VPC endpoint services in the common Availability Zone, `us-east-1a`.

Rules and limitations

To use endpoint services for Gateway Load Balancer endpoints, be aware of the current rules and limitations:

- If an endpoint service is associated with multiple Gateway Load Balancers, then for a specific Availability Zone, a Gateway Load Balancer endpoint establishes a connection with one load balancer only.
- Private DNS names are not supported.
- Availability Zones in your account might not map to the same locations as Availability Zones in another account. For example, your Availability Zone `us-east-1a` might not be the same location as `us-east-1a` for another account. For more information, see Regions and Zones. When you configure an endpoint service, it's configured in the Availability Zones as mapped to your account.

Create a VPC endpoint service configuration for interface endpoints

You can create an endpoint service configuration using the Amazon VPC console or the command line. Before you begin, ensure that you have created one or more Network Load Balancers in your VPC for your service. For more information, see Getting started with Network Load Balancers in the *User Guide for Network Load Balancers*.

In your configuration, you can optionally specify that any interface endpoint connection requests to your service must be manually accepted by you. You can create a notification (p. 52) to receive alerts when there are connection requests. If you do not accept a connection, service consumers cannot access your service.

> **Note**
> Regardless of the acceptance settings, service consumers must also have permissions (p. 48) to create a connection to your service.

After you create an endpoint service configuration, you must add permissions to enable service consumers to create interface endpoints to your service.

Console

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**, **Create Endpoint Service**.
3. For **Associate Load Balancers**, select the Network Load Balancers to associate with the endpoint service.
4. For **Require acceptance for endpoint**, select the check box to accept connection requests to your service manually. If you do not select this option, endpoint connections are automatically accepted.
5. To associate a private DNS name with the service, select **Enable private DNS name**, and then for **Private DNS name**, enter the private DNS name.
6. (Optional) Add or remove a tag.

   [Add a tag] Choose **Add tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Choose the delete button (“x”) to the right of the tag’s Key and Value.
7. Choose **Create service**.

AWS CLI

To create an endpoint service using the AWS CLI

Use the create-vpc-endpoint-service-configuration command and specify one or more ARNs for your Network Load Balancers. You can optionally specify if acceptance is required for connecting to your service and if the service has a private DNS name.

```
aws ec2 create-vpc-endpoint-service-configuration --network-load-balancer-arns
 arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/net/nlb-
vpce/e94221227f1ba532 --acceptance-required --privateDnsName exampleservice.com
```

```
{
    "ServiceConfiguration": {
        "ServiceType": [
            {
                "ServiceType": "Interface"
            }
        ],
        "NetworkLoadBalancerArns": [
            "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/net/nlb-
vpce/e94221227f1ba532"
        ],
        "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-03d5ebb7d9579a2b3",
        "ServiceState": "Available",
        "ServiceId": "vpce-svc-03d5ebb7d9579a2b3",
        "PrivateDnsName: "exampleService.com",
        "AcceptanceRequired": true,
        "AvailabilityZones": [
            "us-east-1d"
        ],
        "BaseEndpointDnsNames": [
            "vpce-svc-03d5ebb7d9579a2b3.us-east-1.vpce.amazonaws.com"
        ]
    }
}
```

AWS Tools for Windows PowerShell

Use New-EC2VpcEndpointServiceConfiguration.

API

Use CreateVpcEndpointServiceConfiguration.

Create a VPC endpoint service configuration for Gateway Load Balancer endpoints

You can create an endpoint service configuration using the Amazon VPC console or the command line. Before you begin, ensure that you have created one or more Gateway Load Balancers in your VPC for your service. For more information, see Getting started with Gateway Load Balancers.

In your configuration, you can optionally specify that any Gateway Load Balancer endpoint connection requests to your service must be manually accepted by you. You can create a notification (p. 52) to receive alerts when there are connection requests. If you do not accept a connection, service consumers cannot access your service.

After you create an endpoint service configuration, you must add permissions (p. 48) to enable service consumers to create a Gateway Load Balancer endpoint to your service.

Console

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**, **Create Endpoint Service**.
3. For **Associate Load Balancers**, select the Gateway Load Balancers to associate with the endpoint service.
4. For **Require acceptance for endpoint**, select the check box to accept connection requests to your service manually. If you do not select this option, endpoint connections are automatically accepted.
5. (Optional) To add a tag, choose **Add tag** and specify the key and value for the tag.
6. Choose **Create service**.

Command line and API

To create an endpoint service using the AWS CLI

Use the create-vpc-endpoint-service-configuration command and specify one or more ARNs for your Gateway Load Balancers. You can optionally specify if acceptance is required for connecting to your service.

```
aws ec2 create-vpc-endpoint-service-configuration --gateway-load-balancer-arns gateway-load-balancer-arn --no-acceptance-required
```

To create an endpoint service using the AWS Tools for Windows PowerShell or API

- New-EC2VpcEndpointServiceConfiguration (Tools for PowerShell)
- CreateVpcEndpointServiceConfiguration (API)

Add and remove permissions for your endpoint service

After you create your endpoint service configuration, you can control which service consumers can create an interface endpoint or Gateway Load Balancer endpoint to connect to your service. Service consumers are IAM principals—IAM users, IAM roles, and AWS accounts. To add or remove permissions for a principal, you need its Amazon Resource Name (ARN).

- For an AWS account (and therefore all principals in the account), the ARN is in the form `arn:aws:iam::aws-account-id:root`.
- For a specific IAM user, the ARN is in the form `arn:aws:iam::aws-account-id:user/user-name`.
- For a specific IAM role, the ARN is in the form `arn:aws:iam::aws-account-id:role/role-name`.

Note
If you set permission to "anyone can access" and you set the acceptance model to "accept all requests," then you've just made your load balancer public. Because it's easy to obtain an AWS

account, there is no practical limitation on who can access your load balancer even though it has no public IP address.

Console

To add or remove 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** and select your endpoint service.
3. Choose **Actions**, **Add principals to allow list**.
4. Specify the ARN for the principal for which to add permissions. To add more principals, choose **Add principal**. To remove a principal, choose the cross icon next to the entry.

   Note
   Specify `*` to add permissions for all principals. This enables all principals in all AWS accounts to create an endpoint to your endpoint service.
5. Choose **Add to principals that are on the allow list**.
6. To remove a principal, select it in the list and choose **Delete**.

AWS CLI

To add permissions for your endpoint service, use the modify-vpc-endpoint-service-permissions command and use the `--add-allowed-principals` parameter to add one or more ARNs for the principals.

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3
 --add-allowed-principals '["arn:aws:iam::123456789012:root"]'
```

To view the permissions you added for your endpoint service, use the describe-vpc-endpoint-service-permissions command.

```
aws ec2 describe-vpc-endpoint-service-permissions --service-id vpce-
svc-03d5ebb7d9579a2b3
```

```
{
    "AllowedPrincipals": [
        {
            "PrincipalType": "Account",
            "Principal": "arn:aws:iam::123456789012:root"
        }
    ]
}
```

To remove permissions for your endpoint service, use the modify-vpc-endpoint-service-permissions command and use the `--remove-allowed-principals` parameter to remove one or more ARNs for the principals.

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3
 --remove-allowed-principals '["arn:aws:iam::123456789012:root"]'
```

AWS Tools for Windows PowerShell

Use Edit-EC2EndpointServicePermission.

API

Use ModifyVpcEndpointServicePermissions.

Change the load balancers and acceptance settings

You can modify your endpoint service configuration by changing the load balancers that are associated with the endpoint service, and by changing whether acceptance is required for requests to connect to your endpoint service.

You cannot disassociate a load balancer if there are endpoints attached to your endpoint service.

Console

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

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoint Services** and select your endpoint service.
3. Choose **Actions**, **Associate/Disassociate Load Balancers**.
4. Select or deselect the load balancers as required, and choose **Save**.

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** and select your endpoint service.
3. Choose **Actions**, **Modify endpoint acceptance setting**.
4. Select or deselect **Require acceptance for endpoint**, and choose **Modify**.

AWS CLI

To change the load balancers for your endpoint service, use the modify-vpc-endpoint-service-configuration command. The following example uses the `--remove-network-load-balancer-arn` parameter to remove a Network Load Balancer.

```
aws ec2 modify-vpc-endpoint-service-configuration --service-
id vpce-svc-09222513e6e77dc86 --remove-network-load-balancer-arn
 arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/net/nlb-
vpce/e94221227f1ba532
```

To change whether acceptance is required, use the modify-vpc-endpoint-service-configuration command and specify `--acceptance-required` or `--no-acceptance-required`.

```
aws ec2 modify-vpc-endpoint-service-configuration --service-id vpce-
svc-09222513e6e77dc86 --no-acceptance-required
```

AWS Tools for Windows PowerShell

Use Edit-EC2VpcEndpointServiceConfiguration.

API

Use ModifyVpcEndpointServiceConfiguration.

Accept and reject endpoint connection requests

After you create an endpoint service, service consumers for which you've added permission can create an interface endpoint or Gateway Load Balancer endpoint to connect to your service. For more information,

see Interface VPC endpoints (AWS PrivateLink) (p. 3) and Gateway Load Balancer endpoints (AWS PrivateLink) (p. 18).

If you specified that acceptance is required for connection requests, you must manually accept or reject endpoint connection requests to your endpoint service. After an endpoint is accepted, it becomes `available`. Be aware that it can take time for a validation status change to be completed and the state to be `available`.

You can reject an endpoint connection after it's in the `available` state.

Console

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** and select your endpoint service.
3. The **Endpoint Connections** tab lists endpoint connections that are currently pending your approval. Select the endpoint, choose **Actions**, and choose **Accept endpoint connection request** to accept the connection or **Reject endpoint connection request** to reject it.

AWS CLI

The view the endpoint connections that are pending acceptance, use the describe-vpc-endpoint-connections command and filter by the `pendingAcceptance` state.

```
aws ec2 describe-vpc-endpoint-connections --filters Name=vpc-endpoint-
state,Values=pendingAcceptance
```

```
{
    "VpcEndpointConnections": [
        {
            "VpcEndpointId": "vpce-0c1308d7312217abc",
            "ServiceId": "vpce-svc-03d5ebb7d9579a2b3",
            "CreationTimestamp": "2017-11-30T10:00:24.350Z",
            "VpcEndpointState": "pendingAcceptance",
            "VpcEndpointOwner": "123456789012"
        }
    ]
}
```

To accept an endpoint connection request, use the accept-vpc-endpoint-connections command and specify the endpoint ID and endpoint service ID.

```
aws ec2 accept-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --vpc-
endpoint-ids vpce-0c1308d7312217abc
```

To reject an endpoint connection request, use the reject-vpc-endpoint-connections command.

```
aws ec2 reject-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --vpc-
endpoint-ids vpce-0c1308d7312217abc
```

AWS Tools for Windows PowerShell

Use Confirm-EC2EndpointConnection and Deny-EC2EndpointConnection.

API

Use AcceptVpcEndpointConnections and RejectVpcEndpointConnections.

Create and manage a notification for an endpoint service

You can create a notification to receive alerts for specific events that occur on the endpoints that are attached to your endpoint service. For example, you can receive an email when an endpoint request is accepted or rejected for your endpoint service. To create a notification, you must associate an Amazon SNS topic with the notification. You can subscribe to the SNS topic to receive an email notification when an endpoint event occurs. For more information, see the Amazon Simple Notification Service Developer Guide.

The Amazon SNS topic that you use for notifications must have a topic policy that allows the Amazon VPC endpoint service to publish notifications on your behalf. Ensure that you include the following statement in your topic policy. For more information, see Managing Access to Your Amazon SNS Topics in the *Amazon Simple Notification Service Developer Guide*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vpce.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:region:account:topic-name"
    }
  ]
}
```

Console

To create a notification for an endpoint service

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoint Services** and select your endpoint service.
3. Choose **Notifications**, **Create Notification**.
4. Choose the ARN for the SNS topic to associate with the notification.
5. For **Events**, select the endpoint events for which to receive notifications.
6. Choose **Create Notification**.

After you create a notification, you can change the SNS topic that's associated with the notification. You can also specify different endpoint events for the notification.

To modify a notification for an endpoint service

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoint Services** and select your endpoint service.
3. Choose **Notifications**, **Actions**, **Modify Notification**.
4. Specify the ARN for the SNS topic and select or deselect the endpoint events as required.
5. Choose **Modify Notification**.

If you no longer need a notification, you can delete it.

To delete a notification

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoint Services** and select your endpoint service.
3. Choose **Notifications**, **Actions**, **Delete Notification**.
4. Choose **Yes, Delete**.

AWS CLI

To create and manage a notification using the AWS CLI

1. To create a notification for an endpoint service, use the create-vpc-endpoint-connection-notification command and specify the ARN of the SNS topic, the events for which to be notified, and the ID of the endpoint service.

```
aws ec2 create-vpc-endpoint-connection-notification --connection-notification-
arn arn:aws:sns:us-east-2:123456789012:VpceNotification --connection-events Connect
 Accept Delete Reject --service-id vpce-svc-1237881c0d25a3abc
```

```
{
    "ConnectionNotification": {
        "ConnectionNotificationState": "Enabled",
        "ConnectionNotificationType": "Topic",
        "ServiceId": "vpce-svc-1237881c0d25a3abc",
        "ConnectionEvents": [
            "Reject",
            "Accept",
            "Delete",
            "Connect"
        ],
        "ConnectionNotificationId": "vpce-nfn-008776de7e03f5abc",
        "ConnectionNotificationArn": "arn:aws:sns:us-
east-2:123456789012:VpceNotification"
    }
}
```

2. To view your notifications, use the describe-vpc-endpoint-connection-notifications command.

```
aws ec2 describe-vpc-endpoint-connection-notifications
```

3. To change the SNS topic or endpoint events for the notification, use the modify-vpc-endpoint-connection-notification command.

```
aws ec2 modify-vpc-endpoint-connection-notification --connection-notification-
id vpce-nfn-008776de7e03f5abc --connection-events Accept Reject --connection-
notification-arn arn:aws:sns:us-east-2:123456789012:mytopic
```

4. To delete a notification, use the delete-vpc-endpoint-connection-notifications command.

```
aws ec2 delete-vpc-endpoint-connection-notifications --connection-notification-
ids vpce-nfn-008776de7e03f5abc
```

AWS Tools for Windows PowerShell

Use New-EC2VpcEndpointConnectionNotification, Get-EC2EndpointConnectionNotification, Edit-EC2VpcEndpointConnectionNotification, and Remove-EC2EndpointConnectionNotification.

API

Use CreateVpcEndpointConnectionNotification, DescribeVpcEndpointConnectionNotifications, ModifyVpcEndpointConnectionNotification, and DeleteVpcEndpointConnectionNotifications.

Add or remove VPC endpoint service tags

Tags provide a way to identify the VPC endpoint service. You can add or remove a tag.

Console

To add or remove a VPC endpoint service tag

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 choose **Actions**, **Add/Edit Tags**.
4. Add or remove a tag.

   [Add a tag] Choose **Create tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Choose the delete button ("x") to the right of the tag's Key and Value.

AWS Tools for Windows PowerShell

Use CreateTags and DeleteTags.

API

Use create-tags and delete-tags.

Delete an endpoint service configuration

You can delete an endpoint service configuration. Deleting the configuration does not delete the application hosted in your VPC or the associated load balancers.

Before you delete the endpoint service configuration, you must reject any `available` or `pending-acceptance` VPC endpoints that are attached to the service. For more information, see Accept and reject endpoint connection requests (p. 50).

Console

To delete an endpoint service configuration using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoint Services** and select the service.
3. Choose **Actions**, **Delete**.
4. Choose **Yes, Delete**.

AWS CLI

To delete an endpoint service configuration using the AWS CLI

- Use the delete-vpc-endpoint-service-configurations command and specify the ID of the service.

```
aws ec2 delete-vpc-endpoint-service-configurations --service-ids vpce-
svc-03d5ebb7d9579a2b3
```

AWS Tools for Windows PowerShell

Use Remove-EC2EndpointServiceConfiguration.

API

Use DeleteVpcEndpointServiceConfigurations.

Identity and access management for VPC endpoints and VPC endpoint services

Use IAM to manage access to VPC endpoints and VPC endpoints services.

Control the use of VPC endpoints

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

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

For information about controlling access to services using VPC endpoints, see the section called “Control access to services” (p. 37).

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 `aws-account-id`). In the following example, you can only create VPC endpoints when the service owner is `amazon`. To use this example, substitute the account ID, the service owner, and the Region (unless you are in the `us-east-1` Region).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:us-east-1:accountId: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. In the following example, you can only create or VPC endpoint service when the Private DNS name is `example.com`. To use this example, substitute the account ID, the Private DNS name, and the Region (unless you are in the `us-east-1` Region).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ModifyVpcEndpointServiceConfiguration",
                "ec2:CreateVpcEndpointServiceConfiguration"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:accountId: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. In the following example, you can only create or VPC endpoint when the service name is `com.amazonaws.us-east-1.s3`. To use this example, substitute the account ID, the service name, and the Region (unless you are in the `us-east-1` Region).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:us-east-1:accountId:vpc-endpoint/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:VpceServiceName": [
                        "com.amazonaws.us-east-1.s3"
                    ]
                }
            }
        }
    ]
}
```

Private DNS names for endpoint services

When you create a VPC endpoint service, we generate endpoint-specific DNS hostnames that you can use to communicate with the service. These names include the VPC endpoint ID, the Availability Zone name and Region Name, for example, vpce-1234-abcdev-us-east-1.vpce-svc-123345.us-east-1.vpce.amazonaws.com. By default, your consumers access the service with that DNS name and usually need to modify the application configuration.

If the endpoint service is for an AWS service, or a service available in the AWS Marketplace, there is a default DNS name. For other services, the service provider can configure a private DNS name so consumers can access the service using an existing DNS name without making changes to their applications. For more information, see *VPC endpoint services (AWS PrivateLink)* (p. 40).

Service providers can specify a private DNS name for a new endpoint service, or an existing endpoint service. To use a private DNS name, enable the feature, and then specify a private DNS name. Before consumers can use the private DNS name, you must verify that you have control of the domain/subdomain. You can initiate domain ownership verification using the Amazon VPC Console or API. After the domain ownership verification completes, consumers access the endpoint by using the private DNS name.

> **Note**
> In order to verify the domain, you need to have a public hosted name, or a public DNS provider. Private DNS names are not supported for endpoint services that you create for Gateway Load Balancer endpoints.

The high-level procedure is as follows:

1. Add a private DNS name. For more information, see the section called “Create a VPC endpoint service configuration for interface endpoints” (p. 46) or the section called “Modify an existing endpoint service private DNS name” (p. 61).
2. Note the **Domain verification value** and **Domain verification name** that you need for the DNS server records. For more information, see the section called “View endpoint service private DNS name configuration” (p. 61).
3. Add a record to the DNS server. For more information, see the section called “VPC endpoint service private DNS name verification” (p. 59).
4. Verify the private DNS name. For more information, see the section called “Manually initiate the endpoint service private DNS name domain verification” (p. 62).

You can manage the verification process by using the Amazon VPC console or the Amazon VPC API.

- the section called “VPC endpoint service private DNS name verification” (p. 59)
- the section called “Modify an existing endpoint service private DNS name” (p. 61)
- the section called “Remove an endpoint service private DNS name” (p. 62)
- the section called “View endpoint service private DNS name configuration” (p. 61)
- Amazon VPC private DNS name domain verification TXT records (p. 63)

Domain name verification considerations

Make note of the following important points about domain ownership verification:

- A consumer can only use the private DNS name to access the endpoint service when the verification status is **verified**.
- If the verification status changes from **verified** to **pendingVerification**, or **failed**, existing consumer connections remain, but any new connection requests are denied.

  Important
  For service providers who are concerned about connections to endpoint services that are no longer in the **verified** state, we recommend that you use DescribeVpcEndpoints to periodically check the verification state. We recommend that you perform this check at least one time per day.
- An endpoint service can only have one private DNS name.
- You can specify a private DNS name for a new endpoint service, or an existing endpoint service.
- You can only use public domain name servers.
- You can use wildcards in domain names, for example, "*.myexampleservice.com".
- You must perform a separate domain ownership verification check for each endpoint service.
- You can verify the domain of a subdomain. For example, you can verify *example.com*, instead of *a.example.com*. As specified in RFC 1034, each DNS label can have up to 63 characters and the whole domain name must not exceed a total length of 255 characters.

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

VPC endpoint service private DNS name verification

Your domain is associated with a set of Domain Name System (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. Each TXT record consists of a name and a value.

When you initiate domain ownership verification using the Amazon VPC Console or API, we give you the name and value to use for the TXT record. For example, if your domain is myexampleservice.com, the TXT record settings that we generate will look similar to the following example:

Endpoint private DNS name TXT record

Domain verification name	Type	Domain verification value
_vpce:aksldja21i1	TXT	vpce:asjdakjshd78126eu21

Add a TXT record to your domain's DNS server using the specified **Domain verification name** and **Domain verification value**. The domain ownership verification is complete when we detect the existence of the TXT record in your domain's DNS settings.

If your DNS provider does not allow DNS record names to contain underscores, you can omit *_aksldja21i1* from the **Domain verification name**. In that case, for the preceding example, the TXT record name would be myexampleservice.com instead of *_aksldja21i1.myexampleservice.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. This section provides procedures for adding a TXT record to Route 53, and generic procedures that apply to other DNS providers.

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. Choose **Endpoint Services**.
3. Select the endpoint service.
4. On the **Details** tab, note the values shown next to **Domain verification value** and **Domain verification name**.
5. If Route 53 provides the DNS service for the domain that you're verifying, and you're signed in to the AWS Management Console under the same account that you use for Route 53, we give you the option of updating your DNS server immediately from within the Amazon VPC console.

   If you use a different DNS provider, the procedures for updating the DNS records vary depending on which DNS or web hosting provider you use. The following table lists links to the documentation for several common providers. This list isn't exhaustive and inclusion in this list isn't an endorsement or recommendation of any company's products or services. If your provider isn't listed in the table, you can probably use the domain with endpoints.

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

   When verification is complete, the domain's status in the Amazon VPC console changes from **Pending** to **Verified**.
6. You can now use the private domain name for the VPC endpoint service.

If the DNS settings are not correctly updated, the domain status displays a status of **failed** on the **Details** tab. If this happens, complete the steps on the troubleshooting page at the section called "Troubleshoot common domain verification problems" (p. 64). After you verify that your TXT record was created correctly, retry the operation.

Modify an existing endpoint service private DNS name

You can modify the endpoint service private DNS name for a new or existing endpoint service.

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, and then choose **Actions**, **Modify private DNS name**.
4. Select **Enable private DNS name**, and then for **Private DNS name**, enter the private DNS name.
5. Choose **Modify**.

After you update the name, update the entry for the domain on your DNS server. We automatically poll the DNS server to verify that the record exists on the server. DNS record updates can take up to 48 hours to take effect, but they often take effect much sooner. For more information, see the section called "Private DNS name domain verification TXT records" (p. 63) and the section called "VPC endpoint service private DNS name verification" (p. 59).

To modify the endpoint service private DNS name using the AWS CLI or API

- modify-vpc-endpoint-service-configuration
- ModifyVpcEndpointServiceConfiguration

View endpoint service private DNS name configuration

You can view the endpoint service private DNS name for an endpoint service.

Console

To view an endpoint service private DNS name configuration using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoint Services**, and then select the endpoint service.
3. The **Details** tab displays the following information for the private DNS domain ownership check:
   - **Domain verification status**: The verification status.
   - **Domain verification type**: The verification type.
   - **Domain verification value**: The DNS value.
   - **Domain verification name**: The name of the record subdomain.

AWS CLI

Use describe-vpc-endpoint-service-configurations.

API

Use DescribeVpcEndpointServiceConfigurations.

Manually initiate the endpoint service private DNS name domain verification

The service provider must prove that they own the private DNS name domain before consumers can use the private DNS name.

Console

To initiate the verification process of the private DNS name domain 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 then choose **Actions**, **Verify domain owernship for Private DNS Name**.
4. Choose **Verify**.

   If the DNS settings are not correctly updated, the domain will display a status of **failed** on the **Details** tab. If this happens, complete the steps on the troubleshooting page at the section called “Troubleshoot common domain verification problems” (p. 64).

AWS CLI

Use start-vpc-endpoint-service-private-dns-verification.

API

Use StartVpcEndpointServicePrivateDnsVerification.

Remove an endpoint service private DNS name

You can remove the endpoint service private DNS name only after there are no connections to the service.

Console

To remove 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, and then choose **Actions**, **Modify private DNS name**.
4. Clear **Enable private DNS name**, and then clear the **Private DNS name**.
5. Choose **Modify**.

AWS CLI

Use modify-vpc-endpoint-service-configuration.

API

Use ModifyVpcEndpointServiceConfiguration.

Amazon VPC private DNS name domain verification TXT records

Your domain is associated with a set of Domain Name System (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. Each TXT record consists of a name and a value.

When you initiate domain ownership verification using the Amazon VPC Console or API, we give you the name and value to use for the TXT record. For example, if your domain is myexampleservice.com, the TXT record settings that Amazon VPC generates will look similar to the following example:

Endpoint private DNS name TXT record

Domain verification name	Type	Domain verification value
_vpce:aksldja21i1.myexampleservice.com	TXT	vpce:asjdakjshd78126eu21

Add a TXT record to your domain's DNS server using the specified **Domain verification name** and **Domain verification value**. Amazon VPC domain ownership verification is complete when Amazon VPC detects the existence of the TXT record in your domain's DNS settings.

If your DNS provider does not allow DNS record names to contain underscores, you can use the domain name for the **Domain verification name**. In that case, for the preceding example, the TXT record name would be myexampleservice.com.

You can find troubleshooting information and instructions on how to check your domain ownership verification settings in Troubleshoot common private DNS domain verification problems (p. 64).

Amazon Route 53

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. This section provides procedures for adding a TXT record to Route 53, and generic procedures that apply to other DNS providers.

To add a TXT record to the DNS record for your Route 53-managed domain

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. Choose **Endpoint Services**.
3. Select the endpoint service.
4. On the **Details** tab, note the values shown next to **Domain verification value** and **Domain verification name**.
5. In the Amazon Route 53 Console, create a record for your hosted zone. For information about how to create a record, see Creating records by using the Amazon Route 53 console in the *Amazon Route 53 Developer Guide*. Use the the following values:
   - For **Record type**, choose **TXT**.
   - For **TTL (Seconds)**, enter **1800**.
   - For **Routing policy**, choose **Simple routing**.
   - For **Value/Route traffic to**, enter the **Domain verification value** from the Amazon VPC console.
6. On the **Details** tab of the **Endpoint Services** page in the Amazon VPC console, check the value in the **Domain verification status** column next for the endpoint. If the status is "pending verification," wait a few minutes, and then choose **refresh**. Repeat this process until the value

in the status column is "verified". You can manually start the verification process. For more information, see the section called “Manually initiate the endpoint service private DNS name domain verification” (p. 62).

Generic procedures for other DNS providers

The procedures for adding TXT records to the DNS configurations vary from provider to provider. For specific steps, consult your DNS provider's documentation. The procedure in this section gives a basic overview of the steps you take when adding a TXT record to the DNS configuration for your domain.

To add a TXT record to your domain's DNS server (general procedure)

1. Go to your DNS provider's website. If you aren't sure which DNS provider serves your domain, you can look it up by using a free Whois service.
2. On the provider's website, sign in to your account.
3. Find the page for updating your domain's DNS records. This page often has a name such as DNS Records, DNS Zone File, or Advanced DNS. If you're unsure, consult the provider's documentation.
4. Add a TXT record with the name and value provided by AWS.

   Important
   Some DNS providers automatically append the domain name to the end of DNS records. Adding a record that already contains the domain name (such as *_pmBGN/7Mjnf.example.com*) might result in the duplication of the domain name (such as *_pmBGN/7Mjnfexample.com.example.com*). To avoid duplication of the domain name, add a period to the end of the domain name in the DNS record. This will indicate to your DNS provider that the record name is fully qualified (that is, no longer relative to the domain name), and will prevent the DNS provider from appending an additional domain name.
5. Save your changes. DNS record updates can take up to 48 hours to take effect, but they often take effect much sooner.

Troubleshoot common private DNS domain verification problems

To verify an endpoint service private DNS domain name with Amazon VPC, you initiate the process using either the Amazon VPC console or the API. This section contains information that can help you resolve issues with the verification process.

Common domain verification problems

If you attempt to verify a domain and you encounter problems, review the possible causes and solutions below.

- You're attempting to verify a domain that you don't own. You can't verify a domain unless you own it.
- Your DNS provider doesn't allow underscores in TXT record names. Some DNS providers don't allow you to include the underscore character in the DNS record names for your domain. If this is true for your provider, you can omit *_amazovpc* from the name of the TXT record.
- 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. For example, if you create a record where the attribute name is *_amazonvpc.example.com*, the provider might append the domain name, resulting in *_amazonvpc.example.com.example.com*). To avoid duplication of

the domain name, add a period to the end of the domain name when you create the TXT record. This step tells your DNS provider that it isn't necessary to append the domain name to the TXT record.

- Your DNS provider modified the DNS record value. Some providers automatically modify DNS record values to use only lowercase letters. We only verify your domain when it detects a verification record for which the attribute value exactly matches the value that we provided when you started the domain ownership verification process. If the DNS provider for your domain changes your TXT record values to use only lowercase letters, contact the DNS provider for additional assistance.
- You want to verify the same domain multiple times. You might need to verify your domain more than once because you're sending in different Regions, or because you're using the same domain to send from multiple AWS accounts. If your DNS provider doesn't allow you to have more than one TXT record with the same attribute name, you might still be able to verify two domains. If your DNS provider allows it, you can assign multiple attribute values to the same TXT record. For example, if your DNS is managed by Amazon Route 53, you can set up multiple values for the same TXT record by completing the following steps:

  1. In the Route 53 console, choose the TXT record that you created when you verified your domain in the first Region.
  2. In the **Value** box, 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 another time with the value removed from the attribute name. For example, you verify with “_asnbcdasd”, and then with "asnbcdasd”. The downside of this solution is that you can only verify the same domain two times.

How to check domain verification settings

You can verify that your private DNS name domain ownership verification TXT record is published correctly to your DNS server by using the following procedure. This procedure uses the nslookup tool, which is available for Windows and Linux. On Linux, you can also use dig.

The commands in these instructions are executed on Windows 7, and the example domain we use is *example.com*.

In this procedure, you first find the DNS servers that serve your domain, and then query those servers to view the TXT records. You query the DNS servers that serve your domain because those servers contain the most up-to-date information for your domain, which can take time to propagate to other DNS servers.

To verify that your domain ownership verification TXT record is published to your DNS server

1. Find the name servers for your domain by taking the following steps.

   a. Go to the command line. To get to the command line on Windows 7, choose **Start** and then enter **cmd**. On Linux-based operating systems, open a terminal window.

   b. At the command prompt, enter the following, where *<domain>* is your domain.

   ```
   nslookup -type=NS <domain>
   ```

   For example, if your domain was *example.com*, the command would look like the following.

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

   The command's output will list the name servers that serve your domain. You will query one of these servers in the next step.

2. Verify that the TXT record is correctly published by taking the following steps.

   a. At the command prompt, enter the following, where *<domain>* is your domain, and *<name server>* is one of the name servers you found in step 1.

   ```
   nslookup -type=TXT  _aksldja21i1.<domain> <name server>
   ```

   In our *_aksldja21i1.example.com* example, if a name server that we found in step 1 was called *ns1.name-server.net*, we would enter the following.

   ```
   nslookup -type=TXT  _aksldja21i1.example.com ns1.name-server.net
   ```

   b. In the output of the command, verify that the string that follows `text =` matches the TXT value you see when you choose the domain in the Identities list of the Amazon VPC console.

   In our example, we are looking for a TXT record under *_aksldja21i1.example.com* with a value of `asjdakjshd78126eu21`. If the record is correctly published, we would expect the command to have the following output.

   ```
   _aksldja21i1.example.com text = "asjdakjshd78126eu21"
   ```

AWS services that integrate with AWS PrivateLink

The following services integrate with AWS PrivateLink. You can create an interface endpoint (p. 3) to connect to these services.

The **VPC endpoint policies** column displays " No", when the service integrates with AWS PrivateLink, but does not support VPC endpoint policies. Choose the "Yes" link to see the documentation for services that support VPC endpoint policies.

AWS service	VPC endpoint policies
Amazon API Gateway	Yes
Amazon AppStream 2.0	No
AWS App Mesh	No
Application Auto Scaling	Yes
Amazon Athena	Yes
AWS Audit Manager	Yes
Amazon Aurora	Yes
AWS Auto Scaling	Yes
AWS Certificate Manager Private Certificate Authority	Yes
Amazon Cloud Directory	Yes
AWS CloudFormation	No
AWS CloudHSM	Yes
AWS CloudTrail	No
Amazon CloudWatch	Yes
Amazon CloudWatch Events	Yes
Amazon CloudWatch Logs	Yes
AWS CodeArtifact	Yes
AWS CodeBuild	Yes
AWS CodeCommit	Yes

AWS service	VPC endpoint policies
AWS CodeDeploy	Yes
Amazon CodeGuru Profiler	No
Amazon CodeGuru Reviewer	No
AWS CodePipeline	No
Amazon Comprehend	Yes
AWS Config	No
Amazon Connect Customer Profiles	Yes
AWS Data Exchange	Yes
AWS DataSync	No
AWS Device Farm	No
Amazon EBS direct APIs	No
Amazon EC2	Yes
EC2 Image Builder	Yes
Amazon EC2 Auto Scaling	Yes
AWS Elastic Beanstalk	Yes
Amazon Elastic File System	Yes
Elastic Load Balancing	Yes
Amazon Elastic Container Registry	Yes
Amazon Elastic Container Service	Yes
Amazon EMR	Yes
Amazon EventBridge	Yes
AWS Fault Injection Simulator	Yes
Amazon Fraud Detector	Yes
AWS Glue	No
AWS IoT SiteWise	No
Amazon Kendra	Yes

AWS service	VPC endpoint policies
AWS Key Management Service	Yes
Amazon Keyspaces (for Apache Cassandra)	Yes
Amazon Kinesis Data Firehose	Yes
Amazon Kinesis Data Streams	Yes
AWS Lambda	Yes
AWS License Manager	Yes
Amazon Lookout for Equipment	Yes
Amazon Managed Blockchain	No
Amazon QLDB	Yes
Amazon RDS	Yes
Amazon RDS Data API	Yes
Amazon Redshift	Yes
Amazon Rekognition	Yes
Amazon S3	Yes
Amazon SageMaker and Amazon SageMaker Runtime	Yes
Amazon SageMaker Notebook	Yes
AWS Secrets Manager	Yes
AWS Security Token Service	Yes
AWS Server Migration Service	No
AWS Service Catalog	No
Amazon SES	No
Amazon SNS	Yes
Amazon SQS	Yes
AWS Step Functions	Yes
AWS Systems Manager	Yes
AWS Storage Gateway	No

AWS service	VPC endpoint policies
Amazon Textract	Yes
Amazon Transcribe	Yes
Amazon Transcribe Medical	Yes
AWS Transfer for SFTP	No
Amazon WorkSpaces	No
Endpoint services (p. 40) hosted by other AWS accounts	No
Supported AWS Marketplace Partner services	No

View available AWS service names

You can use the describe-vpc-endpoint-services command to view the service names that support VPC endpoints.

You can run the following command to get a list of the service names for gateway or interface endpoints. The `--query` option limits the output to the service names

```
aws ec2 describe-vpc-endpoint-services --filter Name=service-type,Values=service-type --
query ServiceNames
```

The following example displays the services that support interface endpoints.

```
aws ec2 describe-vpc-endpoint-services --filter Name=service-type,Values=Interface --query
 ServiceNames
```

Output

```
    "aws.sagemaker.us-east-1.notebook",
    "aws.sagemaker.us-east-1.studio",
    "com.amazonaws.us-east-1.access-analyzer",
    "com.amazonaws.us-east-1.acm-pca",
    "com.amazonaws.us-east-1.airflow.api",
    "com.amazonaws.us-east-1.airflow.env",
    "com.amazonaws.us-east-1.airflow.ops",
    "com.amazonaws.us-east-1.application-autoscaling",
    "com.amazonaws.us-east-1.appmesh-envoy-management",
    "com.amazonaws.us-east-1.appstream.api",
    "com.amazonaws.us-east-1.appstream.streaming",
    "com.amazonaws.us-east-1.aps-workspaces",
    "com.amazonaws.us-east-1.athena",
    "com.amazonaws.us-east-1.auditmanager",
    "com.amazonaws.us-east-1.autoscaling",
    "com.amazonaws.us-east-1.autoscaling-plans",
    "com.amazonaws.us-east-1.awsconnector",
...
```

After you have the service name, you can view detailed information by running the following command.

```
aws ec2 describe-vpc-endpoint-services --filter "Name=service-type,Values=service-type"
 Name=service-name,Values=service-name
```

The following example displays the the information about the Amazon S3 interface endpoint that has a service name of "com.amazonaws.us-east-1.s3 "

```
aws ec2 describe-vpc-endpoint-services --filter "Name=service-type,Values=Interface"
 Name=service-name,Values=com.amazonaws.us-east-1.s3
```

Output

```
{
    "ServiceDetails": [
        {
            "ServiceName": "com.amazonaws.us-east-1.s3",
            "ServiceId": "vpce-svc-081d84efcdEXAMPLE",
            "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": [
                "s3.us-east-1.vpce.amazonaws.com"
            ],
            "VpcEndpointPolicySupported": true,
            "AcceptanceRequired": false,
            "ManagesVpcEndpoints": false,
            "Tags": []
        }
    ],
    "ServiceNames": [
        "com.amazonaws.us-east-1.s3"
    ]
}
```

AWS PrivateLink quotas

The following tables list the quotas, formerly referred to as limits, for AWS PrivateLink resources per Region for your account. Unless indicated otherwise, you can request an increase for these quotas.

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

Resource	Default	Comments
Gateway VPC endpoints per Region	20	You cannot have more than 255 gateway endpoints per VPC.
Interface and Gateway Load Balancer endpoints per VPC	50	This is the combined quota for the maximum number of interface endpoints and Gateway Load Balancer endpoints in a VPC. To increase this quota, contact AWS Support.
VPC endpoint policy size	20,480 characters (including white space)	This quota cannot be increased.

The following maximum transmission unit (MTU) rules apply to traffic that passes through a VPC endpoint.

- The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed through the 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.
- The VPC endpoint does not generate the FRAG_NEEDEDICMP packet, so Path MTU Discovery (PMTUD) is not supported.
- The VPC endpoint enforces Maximum Segment Size (MSS) clamping for all packets. For more information, see RFC879.