Amazon Virtual Private Cloud

AWS PrivateLink

Amazon Virtual Private Cloud: AWS PrivateLink

Table of Contents

What is AWS PrivateLink? 1
- Use cases 1
- Work with VPC endpoints 1
- Pricing 2
- Concepts 2
  - Architecture diagram 2
  - Service providers 3
  - Service consumers 4
  - Private hosted zones 5

Access AWS services 6
- Overview 6
- DNS hostnames 8
- Services that integrate 9
  - View available AWS service names 16
- Create an interface endpoint 17
  - Considerations 17
  - Prerequisites 18
  - Create a VPC endpoint 18
  - Test the VPC endpoint 19
- Configure an interface endpoint 19
  - Add or remove subnets 20
  - Associate security groups 20
  - Edit the VPC endpoint policy 20
  - Enable private DNS names 21
  - Manage tags 21
- Receive alerts for interface endpoint events 22
- Delete an interface endpoint 23
- Gateway endpoints 23
  - Overview 23
  - Routing 24
  - Endpoints for Amazon S3 25
  - Endpoints for DynamoDB 30

Access SaaS products 36
- Overview 36
- Create an interface endpoint 37
- Access the product 37

Access virtual appliances 38
- Overview 38
- Routing 39
- Create a Gateway Load Balancer endpoint service 40
  - Considerations 40
  - Prerequisites 40
  - Create the endpoint service 40
  - Make your endpoint service available 41
- Create a Gateway Load Balancer endpoint 41
  - Considerations 42
  - Prerequisites 42
  - Create the endpoint 42
  - Configure routing 43
  - Manage tags 43
  - Delete the endpoint 44

Share your services 45
- Overview 45
- DNS hostnames 46

IP address types 47
Create an endpoint service 48
Considerations 48
Prerequisites 48
Create an endpoint service 49
Make your endpoint service available to service consumers 49
Configure an endpoint service 51
Manage permissions 51
Accept or reject connection requests 52
Change the load balancer association 53
Associate a private DNS name 53
Modify the supported IP address types 54
Manage tags 54
Manage DNS names 55
Domain ownership verification 55
Get the name and value 56
Add a TXT record to your domain's DNS server 56
Check whether the TXT record is published 57
Troubleshoot domain verification issues 58
Receive alerts for endpoint service events 58
Delete an endpoint service 59
Identity and access management 60
Control access to services 62
VPC endpoint policies 62
Principals for gateway endpoints 63
Update a VPC endpoint policy 63
CloudWatch metrics 64
Endpoint metrics and dimensions 64
Endpoint service metrics and dimensions 66
View the CloudWatch metrics 68
Quotas 69
Document history 70

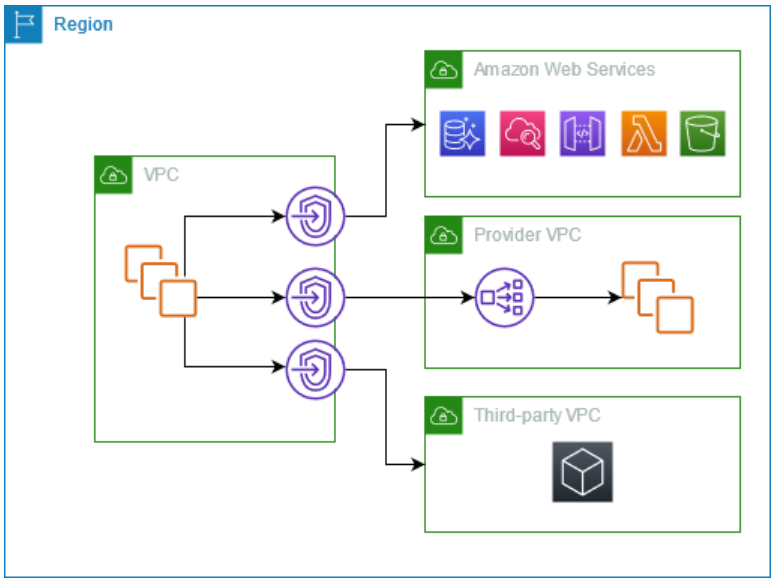
What is AWS PrivateLink?

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

Use cases

You can create VPC endpoints to connect resources in your VPC to services that integrate with AWS PrivateLink. You can create your own VPC endpoint service, powered by AWS PrivateLink, to enable other AWS customers to access your service. For more information, see the section called “Concepts” (p. 2).

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



Learn more

- the section called “Concepts” (p. 2)
- Access AWS services (p. 6)
- Access SaaS products (p. 36)
- Access virtual appliances (p. 38)
- Share your services (p. 45)

Work with VPC endpoints

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

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

Pricing

For information about the pricing for VPC endpoints, see AWS PrivateLink Pricing.

AWS PrivateLink concepts

You can use Amazon VPC to define a virtual private cloud (VPC), which is a logically isolated virtual network. You can launch AWS resources in your VPC. You can provide connectivity to resources outside your VPC to the resources in your VPC using features such as internet gateways, NAT devices, and VPN connections. Alternatively, you can use AWS PrivateLink to connect the resources in your VPC to services using private IP addresses, as if those services were hosted directly in your VPC.

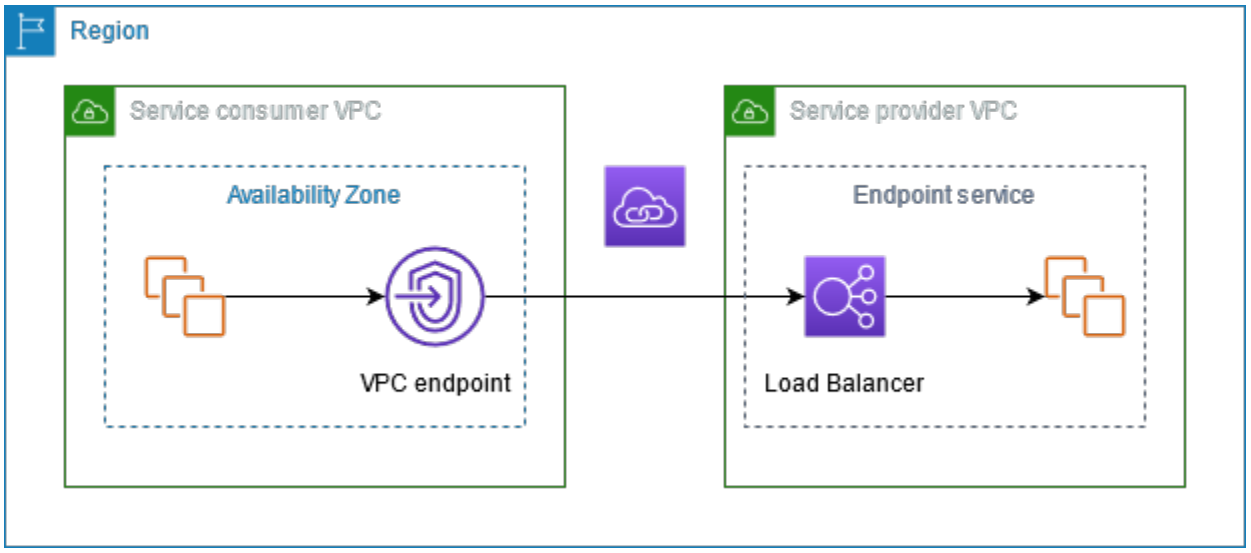
The following are important concepts to understand as you get started using AWS PrivateLink.

Concepts

- Architecture diagram (p. 2)
- Service providers (p. 3)
- Service consumers (p. 4)
- Private hosted zones (p. 5)

Architecture diagram

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



Service providers

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

Endpoint services

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

By default, your endpoint service is not available to service consumers. You must add permissions that allow specific AWS principals (AWS accounts, IAM users, and IAM roles) to connect to your endpoint service.

Service names

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

Service states

The following are the possible states for an endpoint service:

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

Service consumers

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

VPC endpoints

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

- `Interface` - Create an *interface endpoint* to send traffic to endpoint services that use a Network Load Balancer to distribute traffic. Traffic destined for the endpoint service is resolved using DNS.
- `GatewayLoadBalancer` - Create a *Gateway Load Balancer endpoint* to send traffic to a fleet of virtual appliances using private IP addresses. You route traffic from your VPC to the Gateway Load Balancer endpoint using route tables. The Gateway Load Balancer distributes traffic to the virtual appliances and can scale with demand.
- `Gateway` - Create a *gateway endpoint* to send traffic to Amazon S3 or DynamoDB using private IP addresses. You route traffic from your VPC to the gateway endpoint using route tables. Gateway endpoints do not enable AWS PrivateLink.

Endpoint network interfaces

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

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

Endpoint policies

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

Endpoint states

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

The following are the possible states for a VPC endpoint:

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

- `Rejected` - The service provider rejected the connection request. The service provider can also reject a connection after it is available for use.
- `Expired` - The connection request expired.
- `Failed` - The VPC endpoint could not be made available.
- `Deleting` - The service consumer deleted the VPC endpoint and deletion is in progress.
- `Deleted` - The VPC endpoint is deleted.

Private hosted zones

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

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

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

Access AWS services through AWS PrivateLink

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

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

Contents

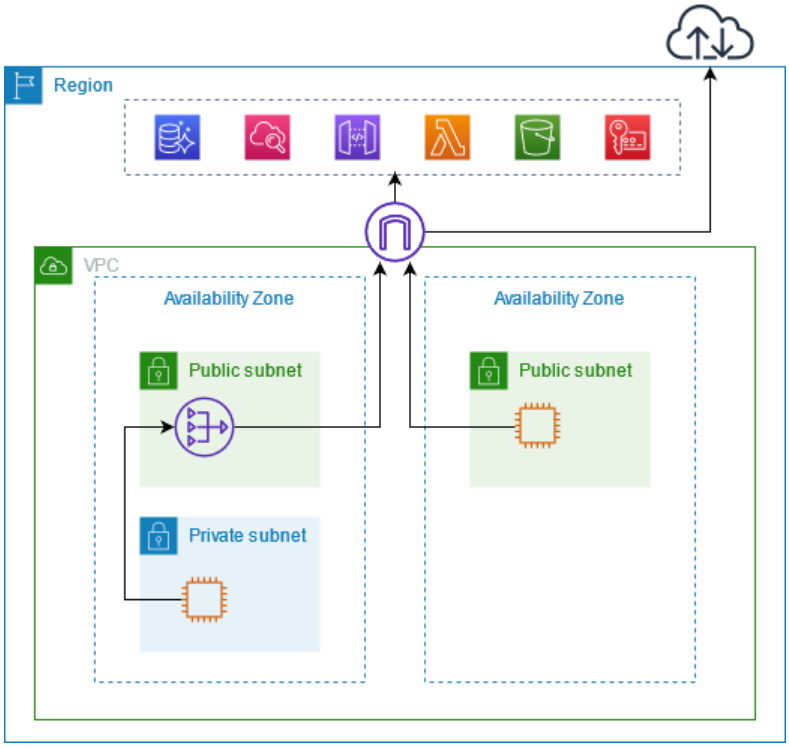
- Overview (p. 6)
- DNS hostnames (p. 8)
- AWS services that integrate with AWS PrivateLink (p. 9)
- Access an AWS service using an interface VPC endpoint (p. 17)
- Configure an interface endpoint (p. 19)
- Receive alerts for interface endpoint events (p. 22)
- Delete an interface endpoint (p. 23)
- Gateway endpoints (p. 23)

Overview

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

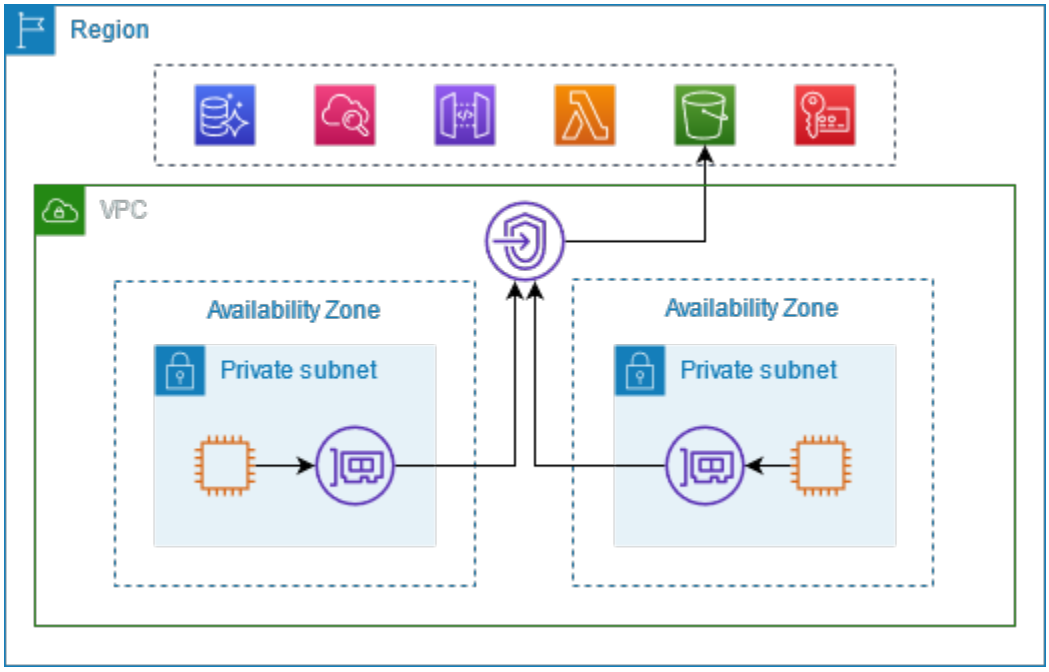
Access through public service endpoints

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



Connect through AWS PrivateLink

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



DNS hostnames

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

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

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

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

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

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

The zonal DNS names have the following syntax:

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

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

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

The following describe-vpc-endpoints command displays the DNS entries for an interface endpoint.

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

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

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

```
            "HostedZoneId": "ZC8PG0KIFKBRI"
        },
        {
            "DnsName": "vpce-099deb00b40f00e22-lj2wisx3-us-east-2b.monitoring.us-
east-2.vpce.amazonaws.com",
            "HostedZoneId": "ZC8PG0KIFKBRI"
        },
        {
            "DnsName": "monitoring.us-east-2.amazonaws.com",
            "HostedZoneId": "Z06320943MMOWYG6MAVL9"
        }
    ]
]
```

AWS services that integrate with AWS PrivateLink

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

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

AWS service	VPC endpoint policies	Service name
Access Analyzer	Yes	com.amazonaws.*region*.access-analyzer
AWS Account Management	Yes	com.amazonaws.*region*.account
Amazon API Gateway	Yes	com.amazonaws.*region*.execute-api
AWS App Mesh	No	com.amazonaws.*region*.appmesh-envoy-management
AWS App Runner	Yes	com.amazonaws.*region*.apprunner
Application Auto Scaling	Yes	com.amazonaws.*region*.application-autoscaling
AWS Application Migration Service	Yes	com.amazonaws.*region*.mgn
Amazon AppStream 2.0	No	com.amazonaws.*region*.appstream.api
		com.amazonaws.*region*.appstream.streaming
Amazon Athena	Yes	com.amazonaws.*region*.athena
AWS Audit Manager	Yes	com.amazonaws.*region*.auditmanager
Amazon Aurora	Yes	com.amazonaws.*region*.rds
AWS Auto Scaling	Yes	com.amazonaws.*region*.autoscaling-plans
AWS Backup	Yes	com.amazonaws.*region*.backup

AWS service	VPC endpoint policies	Service name
AWS Batch	Yes	com.amazonaws.*region*.batch
AWS Billing Conductor	Yes	com.amazonaws.*region*.billingconductor
Amazon Braket	Yes	com.amazonaws.*region*.braket
AWS Certificate Manager Private Certificate Authority	Yes	com.amazonaws.*region*.acm-pca
Amazon Cloud Directory	Yes	com.amazonaws.*region*.clouddirectory
AWS CloudFormation	Yes	com.amazonaws.*region*.cloudformation
AWS CloudHSM	Yes	com.amazonaws.*region*.cloudhsmv2
AWS CloudTrail	No	com.amazonaws.*region*.cloudtrail
Amazon CloudWatch	Yes	com.amazonaws.*region*.evidently
		com.amazonaws.*region*.evidently-dataplane
		com.amazonaws.*region*.monitoring
		com.amazonaws.*region*.rum
		com.amazonaws.*region*.rum-dataplane
		com.amazonaws.*region*.synthetics
Amazon CloudWatch Events	Yes	com.amazonaws.*region*.events
Amazon CloudWatch Logs	Yes	com.amazonaws.*region*.logs
AWS CodeArtifact	Yes	com.amazonaws.*region*.codeartifact.api
		com.amazonaws.*region*.codeartifact.repositories
AWS CodeBuild	Yes	com.amazonaws.*region*.codebuild
		com.amazonaws.*region*.codebuild-fips
AWS CodeCommit	Yes	com.amazonaws.*region*.codecommit
		com.amazonaws.*region*.codecommit-fips
		com.amazonaws.*region*.git-codecommit
		com.amazonaws.*region*.git-codecommit-fips
AWS CodeDeploy	Yes	com.amazonaws.*region*.codedeploy
		com.amazonaws.*region*.codedeploy-commands-secure
Amazon CodeGuru Profiler	No	com.amazonaws.*region*.codeguru-profiler

AWS service	VPC endpoint policies	Service name
Amazon CodeGuru Reviewer	No	com.amazonaws.*region*.codeguru-reviewer
AWS CodePipeline	No	com.amazonaws.*region*.codepipeline
AWS CodeStar Connections	Yes	com.amazonaws.*region*.codestar-connections.api
Amazon Comprehend	Yes	com.amazonaws.*region*.comprehend
Amazon Comprehend Medical	Yes	com.amazonaws.*region*.comprehendmedical
AWS Config	Yes	com.amazonaws.*region*.config
Amazon Connect	Yes	com.amazonaws.*region*.app-integrations
		com.amazonaws.*region*.profile
		com.amazonaws.*region*.voiceid
		com.amazonaws.*region*.wisdom
AWS Data Exchange	Yes	com.amazonaws.*region*.dataexchange
AWS Database Migration Service	Yes	com.amazonaws.*region*.dms
		com.amazonaws.*region*.dms-fips
AWS DataSync	No	com.amazonaws.*region*.datasync
AWS Device Farm	No	
Amazon DevOps Guru	Yes	com.amazonaws.*region*.devops-guru
Amazon EBS direct APIs	No	com.amazonaws.*region*.ebs
Amazon EC2	Yes	com.amazonaws.*region*.ec2
Amazon EC2 Auto Scaling	Yes	com.amazonaws.*region*.autoscaling
EC2 Image Builder	Yes	com.amazonaws.*region*.imagebuilder
Amazon ECR	Yes	com.amazonaws.*region*.ecr.api
		com.amazonaws.*region*.ecr.dkr
Amazon ECS	Yes	com.amazonaws.*region*.ecs
		com.amazonaws.*region*.ecs-agent
		com.amazonaws.*region*.ecs-telemetry
AWS Elastic Beanstalk	Yes	com.amazonaws.*region*.elasticbeanstalk
		com.amazonaws.*region*.elasticbeanstalk-health
AWS Elastic Disaster Recovery	Yes	com.amazonaws.*region*.drs

AWS service	VPC endpoint policies	Service name
Amazon Elastic File System	Yes	com.amazonaws.*region*.elasticfilesystem
		com.amazonaws.*region*.elasticfilesystem-fips
Amazon Elastic Inference	No	com.amazonaws.*region*.elastic-inference.runtime
Elastic Load Balancing	Yes	com.amazonaws.*region*.elasticloadbalancing
Amazon ElastiCache	Yes	com.amazonaws.*region*.elasticache
Amazon EMR	Yes	com.amazonaws.*region*.elasticmapreduce
Amazon EMR on EKS	Yes	com.amazonaws.*region*.emr-containers
Amazon EventBridge	Yes	com.amazonaws.*region*.events
AWS Fault Injection Simulator	Yes	com.amazonaws.*region*.fis
Amazon FinSpace	Yes	com.amazonaws.*region*.finspace
		com.amazonaws.*region*.finspace-api
Amazon Forecast	Yes	com.amazonaws.*region*.forecast
		com.amazonaws.*region*.forecastquery
		com.amazonaws.*region*.forecast-fips
		com.amazonaws.*region*.forecastquery-fips
Amazon Fraud Detector	Yes	com.amazonaws.*region*.frauddetector
Amazon FSx	Yes	com.amazonaws.*region*.fsx
		com.amazonaws.*region*.fsx-fips
AWS Glue	Yes	com.amazonaws.*region*.glue
AWS Glue DataBrew	Yes	com.amazonaws.*region*.databrew
Amazon Managed Grafana	Yes	com.amazonaws.*region*.grafana
AWS Ground Station	Yes	com.amazonaws.*region*.groundstation
Amazon HealthLake	Yes	com.amazonaws.*region*.healthlake
Amazon Inspector	Yes	com.amazonaws.*region*.inspector2
AWS IoT Core	No	com.amazonaws.*region*.iot.data
AWS IoT Core for LoRaWAN	No	com.amazonaws.*region*.iotwireless.api
		com.amazonaws.*region*.lorawan.cups
		com.amazonaws.*region*.lorawan.lns

AWS service	VPC endpoint policies	Service name
AWS IoT Greengrass	Yes	com.amazonaws.*region*.greengrass
AWS IoT SiteWise	No	com.amazonaws.*region*.iotsitewise.api
		com.amazonaws.*region*.iotsitewise.data
AWS IoT TwinMaker	Yes	com.amazonaws.*region*.iottwinmaker.api
		com.amazonaws.*region*.iottwinmaker.data
Amazon Kendra	Yes	com.amazonaws.*region*.kendra
AWS Key Management Service	Yes	com.amazonaws.*region*.kms
Amazon Keyspaces (for Apache Cassandra)	Yes	com.amazonaws.*region*.cassandra
		com.amazonaws.*region*.cassandra-fips
Amazon Kinesis Data Firehose	Yes	com.amazonaws.*region*.kinesis-firehose
Amazon Kinesis Data Streams	Yes	com.amazonaws.*region*.kinesis-streams
AWS Lake Formation	Yes	com.amazonaws.*region*.lakeformation
AWS Lambda	Yes	com.amazonaws.*region*.lambda
Amazon Lex	Yes	com.amazonaws.*region*.models-v2-lex
		com.amazonaws.*region*.runtime-v2-lex
AWS License Manager	Yes	com.amazonaws.*region*.license-manager
		com.amazonaws.*region*.license-manager-fips
Amazon Lookout for Equipment	Yes	com.amazonaws.*region*.lookoutequipment
Amazon Lookout for Metrics	Yes	com.amazonaws.*region*.lookoutmetrics
Amazon Lookout for Vision	Yes	com.amazonaws.*region*.lookoutvision
Amazon Macie	No	com.amazonaws.*region*.macie2
Amazon Managed Blockchain	No	
Amazon Managed Service for Prometheus	No	com.amazonaws.*region*.aps
		com.amazonaws.*region*.aps-workspaces
Amazon Managed Workflows for Apache Airflow	Yes	com.amazonaws.*region*.airflow.api
		com.amazonaws.*region*.airflow.env
		com.amazonaws.*region*.airflow.ops
Amazon MemoryDB for Redis	Yes	com.amazonaws.*region*.memory-db

AWS service	VPC endpoint policies	Service name
		com.amazonaws.*region*.memorydb-fips
Migration Hub Orchestrator	Yes	com.amazonaws.*region*.migrationhub-orchestrator
Migration Hub Strategy Recommendations	Yes	com.amazonaws.*region*.migrationhub-strategy
Amazon Nimble Studio	Yes	com.amazonaws.*region*.nimble
AWS Proton	Yes	com.amazonaws.*region*.proton
Amazon QLDB	Yes	com.amazonaws.*region*.qldb.session
Amazon RDS	Yes	com.amazonaws.*region*.rds
Amazon RDS Data API	Yes	com.amazonaws.*region*.rds-data
Amazon Redshift	Yes	com.amazonaws.*region*.redshift
		com.amazonaws.*region*.redshift-data
		com.amazonaws.*region*.redshift-fips
Amazon Rekognition	Yes	com.amazonaws.*region*.rekognition
		com.amazonaws.*region*.rekognition-fips
AWS RoboMaker	Yes	com.amazonaws.*region*.robomaker
Amazon S3	Yes	com.amazonaws.*region*.s3
Amazon S3 Multi-Region Access Points	Yes	com.amazonaws.s3-global.accesspoint
Amazon S3 on Outposts	Yes	com.amazonaws.*region*.s3-outposts
Amazon SageMaker	Yes	aws.sagemaker.*region*.notebook
		aws.sagemaker.*region*.studio
		com.amazonaws.*region*.sagemaker.api
		com.amazonaws.*region*.sagemaker.featurestore-runtime
		com.amazonaws.*region*.sagemaker.runtime
		com.amazonaws.*region*.sagemaker.runtime-fips
AWS Secrets Manager	Yes	com.amazonaws.*region*.secretsmanager
AWS Security Hub	Yes	com.amazonaws.*region*.securityhub
AWS Security Token Service	Yes	com.amazonaws.*region*.sts
AWS Server Migration Service	No	com.amazonaws.*region*.awsconnector

AWS service	VPC endpoint policies	Service name
		com.amazonaws.*region*.sms
		com.amazonaws.*region*.sms-fips
AWS Service Catalog	Yes	com.amazonaws.*region*.servicecatalog
		com.amazonaws.*region*.servicecatalog-appregistry
Amazon SES	No	com.amazonaws.*region*.email-smtp
AWS Snow Device Management	Yes	com.amazonaws.*region*.snow-device-management
Amazon SNS	Yes	com.amazonaws.*region*.sns
Amazon SQS	Yes	com.amazonaws.*region*.sqs
AWS Step Functions	Yes	com.amazonaws.*region*.states
		com.amazonaws.*region*.sync-states
AWS Storage Gateway	No	com.amazonaws.*region*.storagegateway
AWS Systems Manager	Yes	com.amazonaws.*region*.ec2messages
		com.amazonaws.*region*.ssm
		com.amazonaws.*region*.ssm-contacts
		com.amazonaws.*region*.ssm-incidents
		com.amazonaws.*region*.ssmmessages
Amazon Textract	Yes	com.amazonaws.*region*.textract
		com.amazonaws.*region*.textract-fips
Amazon Transcribe	Yes	com.amazonaws.*region*.transcribe
		com.amazonaws.*region*.transcribestreaming
Amazon Transcribe Medical	Yes	com.amazonaws.*region*.transcribe
		com.amazonaws.*region*.transcribestreaming
AWS Transfer for SFTP	No	com.amazonaws.*region*.transfer
		com.amazonaws.*region*.transfer.server
Amazon Translate	Yes	com.amazonaws.*region*.translate
Amazon WorkSpaces	Yes	com.amazonaws.*region*.workspaces
AWS X-Ray	Yes	com.amazonaws.*region*.xray

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 possible values for the `service-type` filter are `Interface` and `Gateway`. 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
```

The following is example 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",
...
```

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

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

The following example displays information about the Amazon S3 interface endpoint in the `us-east-1` Region. The `service-type` filter excludes the Amazon S3 gateway endpoint from the output.

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

The following is example output:

```
{
    "ServiceDetails": [
        {
            "ServiceName": "com.amazonaws.us-east-1.s3",
            "ServiceId": "vpce-svc-081d84efcdc7bac15",
            "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"
    ]
}
```

Access an AWS service using an interface VPC endpoint

You can create an interface VPC endpoint to connect to services powered by AWS PrivateLink, including many AWS services.

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

You are billed for hourly usage and data processing charges. For more information, see Interface endpoint pricing.

Contents

- Considerations (p. 17)
- Prerequisites (p. 18)
- Create a VPC endpoint (p. 18)
- Test the VPC endpoint (p. 19)

Considerations

- Interface VPC endpoints support traffic only over TCP.
- AWS services accept connection requests automatically. The service can't initiate requests to resources in your VPC through the VPC endpoint. The endpoint only returns responses to traffic that was initiated by resources in your VPC.
- The DNS names created for VPC endpoints are publicly resolvable. They resolve to the private IP addresses of the endpoint network interfaces for the enabled Availability Zones. The private DNS names are not publicly resolvable.
- By default, each interface endpoint can support a bandwidth of up to 10 Gbps per Availability Zone and automatically scales up to 40 Gbps. If your application needs higher throughput per zone, contact AWS Support.

- There are quotas on your AWS PrivateLink resources. For more information, see AWS PrivateLink quotas (p. 69).

Prerequisites

- Create a private subnet in your VPC and deploy the resources that will access the AWS service using the VPC endpoint in the private subnet.
- To use private DNS, you must enable DNS hostnames and DNS resolution for your VPC. For more information, see View and update DNS attributes in the *Amazon VPC User Guide*.
- The security group for the interface endpoint must allow communication between the endpoint network interface and the resources in your VPC that must communicate with the service. By default, the interface endpoint uses the default security group for the VPC. Alternatively, you can create a security group to control the traffic to the endpoint network interfaces from the resources in the VPC. To ensure that tools such as the AWS CLI can make requests over HTTPS from resources in the VPC to the AWS service, the security group must allow inbound HTTPS traffic.
- If your resources are in a subnet with a network ACL, verify that the network ACL allows traffic between the endpoint network interfaces and the resources in the VPC.

Create a VPC endpoint

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

To create an interface endpoint for an AWS service

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create endpoint**.
4. For **Service category**, choose **AWS services**.
5. For **Service name**, select the service. For more information, see the section called “Services that integrate” (p. 9).
6. For **VPC**, select the VPC from which you'll access the AWS service.
7. To create an interface endpoint for Amazon S3, you must clear **Additional settings**, **Enable DNS name**. This is because Amazon S3 does not support private DNS for interface VPC endpoints.
8. For **Subnets**, select one subnet per Availability Zone from which you'll access the AWS service.
9. For **Security group**, select the security groups to associate with the endpoint network interfaces. The security group rules must allow resources that will use the VPC endpoint to communicate with the AWS service to communicate with the endpoint network interface.
10. For **Policy**, select **Full access** to allow all operations by all principals on all resources over the VPC endpoint. Otherwise, select **Custom** to attach a VPC endpoint policy that controls the permissions that principals have for performing actions on resources over the VPC endpoint. This option is available only if the service supports VPC endpoint policies. For more information, see the section called “VPC endpoint policies” (p. 62).
11. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
12. Choose **Create endpoint**.

To create an interface endpoint using the command line

- create-vpc-endpoint (AWS CLI)

- New-EC2VpcEndpoint (Tools for Windows PowerShell)

Test the VPC endpoint

After you create the VPC endpoint, verify that it's sending requests from your VPC to the AWS service. For this example, we'll demonstrate how to send a request from an EC2 instance in your private subnet to an AWS service, such as Amazon CloudWatch. This requires an EC2 instance in a public subnet from which you can access the instance in the private subnet.

Requirement

Verify that the VPC with the interface VPC endpoint has both a public subnet and a private subnet.

To test the VPC endpoint

1. Launch an EC2 instance into the private subnet. Use an AMI that comes with the AWS CLI preinstalled (for example, an AMI for Amazon Linux 2) and add an IAM role that allows the instance to call the AWS service. For example, for Amazon CloudWatch, attach the **CloudWatchReadOnlyAccess** policy to the IAM role.
2. Launch an EC2 instance into the public subnet and connect to this instance. From the instance in the public subnet, connect to the instance in the private subnet using its private IP address, using the following command.

   ```
   $ ssh ec2-user@10.0.0.23
   ```

3. Confirm that the instance in the private subnet does not have connectivity to the internet by pinging a well-known public server. If there is 0% packet loss, the instance has internet access. If there is 100% packet loss, the instance has no internet access. For example, the following command pings the Amazon website one time.

   ```
   $ ping -c 1 www.amazon.com
   ```

4. Run a describe command for the AWS service from the AWS CLI to confirm connectivity to the service from the instance. For example, for Amazon CloudWatch, run the list-metrics command from the instance in the private subnet.

   ```
   $ aws cloudwatch list-metrics --namespace AWS/EC2
   ```

   If you get a response, even a response with empty results, then you are connected to the service using AWS PrivateLink. If the command times out, verify that the instance has an IAM role that allows access to the AWS service.

Configure an interface endpoint

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

Tasks

- Add or remove subnets (p. 20)
- Associate security groups (p. 20)
- Edit the VPC endpoint policy (p. 20)
- Enable private DNS names (p. 21)

- Manage tags (p. 21)

Add or remove subnets

You can choose one subnet per Availability Zone for your interface endpoint. If you add a subnet, we create and endpoint network interface in the subnet and assign it a private IP address from the IP address range of the subnet. If you remove a subnet, we delete its endpoint network interface.

To change the subnets using the console

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

To change the subnets using the command line

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

Associate security groups

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

To change the security groups using the console

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

To change the security groups using the command line

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

Edit the VPC endpoint policy

You can edit the endpoint policy for a VPC endpoint, which controls access to the endpoint service from the VPC through the endpoint. After you update an endpoint policy, it can take a few minutes for the changes to take effect. For more information, see the section called “VPC endpoint policies” (p. 62).

To change the endpoint policy using the console

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

To change the endpoint policy using the command line

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

Enable private DNS names

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

To change the private DNS names option using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints**.
3. Select the interface endpoint.
4. Choose **Actions**, **Modify private DNS name**.
5. Select or clear **Enable for this endpoint** as required.
6. Choose **Save changes**.

To change the private DNS names option using the command line

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

Manage tags

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

To manage tags using the console

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

7. Choose **Save**.

To manage tags using the command line

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

Receive alerts for interface endpoint events

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

Requirement

Create an Amazon SNS topic for the notifications and subscribe to the topic. Add an access policy to the topic that allows the Amazon VPC endpoint service to publish notifications on your behalf, such as the following. For more information, see How do I edit my Amazon SNS topic's access policy?

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

To create a notification for an interface endpoint using the console

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

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

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

Delete an interface endpoint

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

To delete an interface endpoint using the console

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

To delete an interface endpoint using the command line

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

Gateway endpoints

Gateway endpoints provide reliable connectivity to Amazon S3 and DynamoDB without requiring an internet gateway or a NAT device for your VPC. Gateway endpoints do not enable AWS PrivateLink.

There is no additional charge for using gateway endpoints.

Amazon S3 supports both gateway endpoints and interface endpoints. For a comparison of the two options, see Types of VPC endpoints for Amazon S3 in the *Amazon S3 User Guide*.

Contents

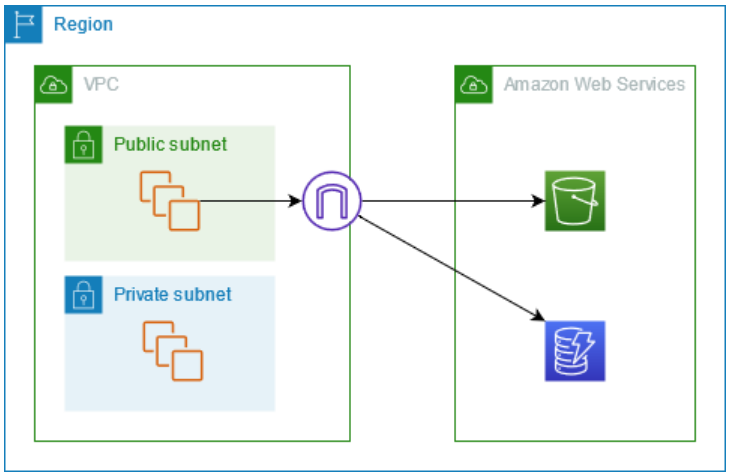
- Overview (p. 23)
- Routing (p. 24)
- Gateway endpoints for Amazon S3 (p. 25)
- Gateway endpoints for Amazon DynamoDB (p. 30)

Overview

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

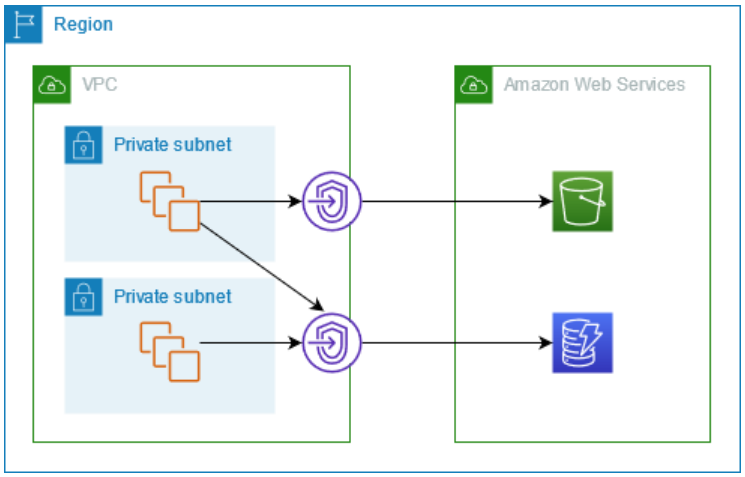
Access through an internet gateway

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



Access through a gateway endpoint

The following diagram shows how instances access Amazon S3 and DynamoDB through a gateway endpoint. Traffic from your VPC to Amazon S3 or DynamoDB is routed to the gateway endpoint. Each subnet route table must have a route that sends traffic destined for the service to the gateway endpoint using the prefix list for the service.



Routing

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

Destination	Target
prefix_list_id	*gateway_endpoint_id*

Considerations

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

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

Gateway endpoints for Amazon S3

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

There is no additional charge for using gateway endpoints.

Amazon S3 supports both gateway endpoints and interface endpoints. For a comparison of the two options, see Types of VPC endpoints for Amazon S3 in the *Amazon S3 User Guide*.

Contents

- Considerations (p. 25)
- Create a gateway endpoint (p. 26)
- Control access using bucket policies (p. 26)
- Associate route tables (p. 28)
- Edit the VPC endpoint policy (p. 29)
- Delete a gateway endpoint (p. 30)

Considerations

- A gateway endpoint is available only in the Region where you created it. Be sure to create your gateway endpoint in the same Region as your S3 buckets.
- If you're using the Amazon DNS servers, you must enable both DNS hostnames and DNS resolution for your VPC. If you're using your own DNS server, ensure that requests to Amazon S3 resolve correctly to the IP addresses maintained by AWS.
- Check whether you are using an AWS service that uses an S3 bucket. If so, ensure that the endpoint policy allows full access to Amazon S3 (the default) or that it allows access to the buckets used by the AWS service. Alternatively, ensure that the requests to Amazon S3 do not originate from a subnet with a route table with an endpoint route for Amazon S3.
- You cannot use an IAM policy or bucket policy to allow access from an VPC IPv4 CIDR range. VPC CIDR blocks can be overlapping or identical, which might lead to unexpected results. Therefore, you can't 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 to 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 route tables to control which instances can access resources in Amazon S3 through the gateway endpoint.
  - Use bucket policies (p. 26) to restrict access to a specific endpoint, VPC, or IP address range.
- The outbound rules for the security group for instances that access Amazon S3 through the gateway endpoint must allow traffic to Amazon S3. You can use the prefix list ID for Amazon S3 as the destination in the outbound rule.
- Gateway endpoints support only IPv4 traffic.
- The source IPv4 addresses from instances in your affected subnets as received by Amazon S3 change from public IPv4 addresses to the private IPv4 addresses in your VPC. An endpoint switches network routes, and disconnects open TCP connections. The previous connections that used public IPv4 addresses are not resumed. We recommend that you do not have any critical tasks running when you create or modify an endpoint; or that you test to ensure that your software can automatically reconnect to Amazon S3 after the connection break.
- Endpoint connections cannot be extended out of a VPC. Resources on the other side of a VPN connection, VPC peering connection, transit gateway, or AWS Direct Connect connection in your VPC cannot use a gateway endpoint to communicate with Amazon S3.
- Your account has a default quota of 20 gateway endpoints per Region, which is adjustable. There is also a limit of 255 gateway endpoints per VPC.

Create a gateway endpoint

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

To create a gateway endpoint using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create endpoint**.
4. For **Service category**, choose **AWS services**.
5. For **Services**, add the filter **Type: Gateway** and select **com.amazonaws.*region*.s3**.
6. For **VPC**, select the VPC in which to create the endpoint.
7. For **Route tables**, select the route tables to be used by the endpoint. We automatically add a route that points traffic destined for the service to the endpoint network interface.
8. For **Policy**, select **Full access** to allow all operations by all principals on all resources over the VPC endpoint. Otherwise, select **Custom** to attach a VPC endpoint policy that controls the permissions that principals have to perform actions on resources over the VPC endpoint.
9. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
10. Choose **Create endpoint**.

To create a gateway endpoint using the command line

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

Control access using bucket policies

You can use bucket policies to control access to buckets from specific endpoints, VPCs, IP address ranges, and AWS accounts.

Example Example: Restrict access to a specific endpoint

You can create a bucket policy that restricts access to a specific endpoint by using the aws:sourceVpce condition key. The following policy denies access to the specified bucket unless the specified gateway endpoint is used. This example assumes that there is also a policy statement that allows the access required for your use cases.

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

Example Example: Restrict access to a specific VPC

You can create a bucket policy that restricts access to specific VPCs by using the aws:sourceVpc condition key. This is useful if you have multiple endpoints configured in the same VPC. The following policy denies access to the specified bucket and its objects that does not come from the specified VPC. This example assumes that there is also a policy statement that allows the access required for your use cases.

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

Example Example: Restrict access to a specific IP address range

You can create a policy that restricts access to specific IP address ranges by using the aws:VpcSourceIp condition key. The following policy denies access to the specified bucket and its objects that does not come from the specified IP address. This example assumes that there is also a policy statement that allows the access required for your use cases.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    {
      "Sid": "Access-to-specific-VPC-CIDR-only",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": ["arn:aws:s3:::bucket_name",
                   "arn:aws:s3:::bucket_name/*"],
      "Condition": {
        "NotIpAddress": {
          "aws:VpcSourceIp": "172.31.0.0/16"
        }
      }
    }
  ]
}
```

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

You can create a policy that restricts access to the S3 buckets in a specific AWS account by using the `s3:ResourceAccount` condition key. The following policy denies access to resources that are not owned by the specified AWS account. This example assumes that there is also a policy statement that allows the access required for your use cases.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    {
      "Sid": "Access-to-bucket-in-specific-account-only",
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "StringNotEquals": {
          "s3:ResourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

Associate route tables

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

To associate route tables using the console

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

3. Select the gateway endpoint.
4. Choose **Actions**, **Manage route tables**.
5. Select or deselect route tables as needed.
6. Choose **Modify route tables**.

To associate route tables using the command line

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

Edit the VPC endpoint policy

You can edit the endpoint policy for a gateway endpoint, which controls access to Amazon S3 from the VPC through the endpoint. The default policy allows full access. For more information, see VPC endpoint policies (p. 62).

To change the endpoint policy using the console

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

The following are example endpoint policies for accessing Amazon S3.

Example Example: Restrict access to a specific bucket

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

```
{
  "Sid": "AccessToSpecificBucket",
  "Effect": "Allow",
  "Principal": "*",
  "Action": [
     "s3:ListBucket",
     "s3:GetObject",
     "s3:PutObject"
  ],
  "Resource": [
     "arn:aws:s3:::bucket_name",
     "arn:aws:s3:::bucket_name/*"
  ]
}
```

Example Example: Restrict access to a specific IAM role

You can create a policy that restricts access to a specific IAM role.

```
{
  "Sid": "Restrict-access-to-specific-IAM-role",
  "Effect": "Allow",
```

```
  "Principal": "*",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "ArnEquals": {
      "aws:PrincipalArn": "arn:aws:iam::111122223333:role/role_name"
    }
  }
}
```

Example Example: Restrict access to users in a specific account

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

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

Delete a gateway endpoint

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

To delete a gateway endpoint using the console

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

To delete a gateway endpoint using the command line

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

Gateway endpoints for Amazon DynamoDB

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

There is no additional charge for using gateway endpoints.

Contents

- Considerations (p. 31)
- Create a gateway endpoint (p. 31)
- Control access using IAM policies (p. 32)
- Associate route tables (p. 33)
- Edit the VPC endpoint policy (p. 33)
- Delete a gateway endpoint (p. 34)

Considerations

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

Create a gateway endpoint

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

To create a gateway endpoint using the console

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

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

To create a gateway endpoint using the command line

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

Control access using IAM policies

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

Example Example: Restrict access to a specific endpoint

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

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

Example Example: Allow access from a specific IAM role

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

```
{
  "Sid": "AllowAccessFromIAMRole",
  "Effect": "Allow",
  "Principal": "*",
  "Action": "*",
  "Resource": "*",
  "Condition": {
    "ArnEquals": {
      "aws:PrincipalArn": "arn:aws:iam::111122223333:role/role_name"
    }
  }
}
```

Example Example: Allows access from a specific account

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

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

Associate route tables

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

To associate route tables using the console

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

To associate route tables using the command line

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

Edit the VPC endpoint policy

You can edit the endpoint policy for a gateway endpoint, which controls access to DynamoDB from the VPC through the endpoint. The default policy allows full access. For more information, see VPC endpoint policies (p. 62).

To change the endpoint policy using the console

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

To modify a gateway endpoint using the command line

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

The following are example endpoint policies for accessing DynamoDB.

Example Example: Allow read-only access

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

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

Example Example: Restrict access to a specific table

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

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

Delete a gateway endpoint

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

To delete a gateway endpoint using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

2. In the navigation pane, choose **Endpoints**.
3. Select the gateway endpoint.
4. Choose **Actions**, **Delete VPC endpoints**.
5. When prompted for confirmation, enter **`delete`**.
6. Choose **Delete**.

To delete an interface endpoint using the command line

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

Access SaaS products through AWS PrivateLink

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

Contents

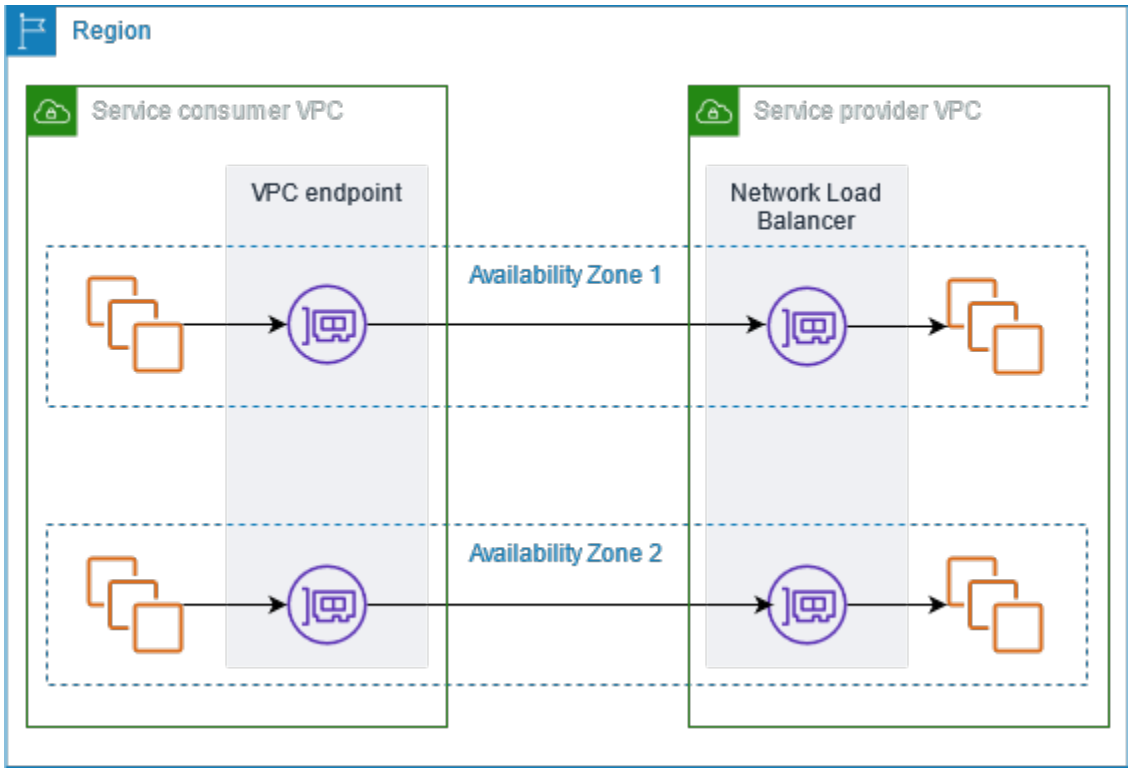
- Overview (p. 36)
- Create an interface endpoint (p. 37)
- Access the product (p. 37)

Overview

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

You can also find SaaS products powered by AWS PrivateLink from AWS Partners. For more information see AWS PrivateLink Partners.

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



Create an interface endpoint

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

Requirement

Subscribe to the service.

To create an interface endpoint to a partner service

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create endpoint**.
4. If you purchased the service from AWS Marketplace, do the following:
   a. For **Service category**, choose **AWS Marketplace services**.
   b. Enter the name of the service.
5. If you subscribed to a service with the AWS Service Ready designation, do the following:
   a. For **Service category**, choose **PrivateLink Ready partner services**.
   b. Enter the name of the service and choose **Verify service**.
6. For **VPC**, select the VPC from which you'll access the product.
7. For **Subnets**, select one subnet per Availability Zone from which you'll access the product.
8. For **Security group**, select the security groups to associate with the endpoint network interfaces. The security group rules must allow traffic between the resources in the VPC and the endpoint network interfaces.
9. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
10. Choose **Create endpoint**.

To configure an interface endpoint

For information about configuring your interface endpoint, see the section called “Configure an interface endpoint” (p. 19).

Access the product

Access the product using the private DNS name provided for you.

If the endpoint service supports it, you can add a VPC endpoint policy for your interface endpoint, which controls access to the endpoint service from the VPC through the endpoint. The default policy allows full access. For more information, see the section called “VPC endpoint policies” (p. 62).

Access virtual appliances through AWS PrivateLink

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

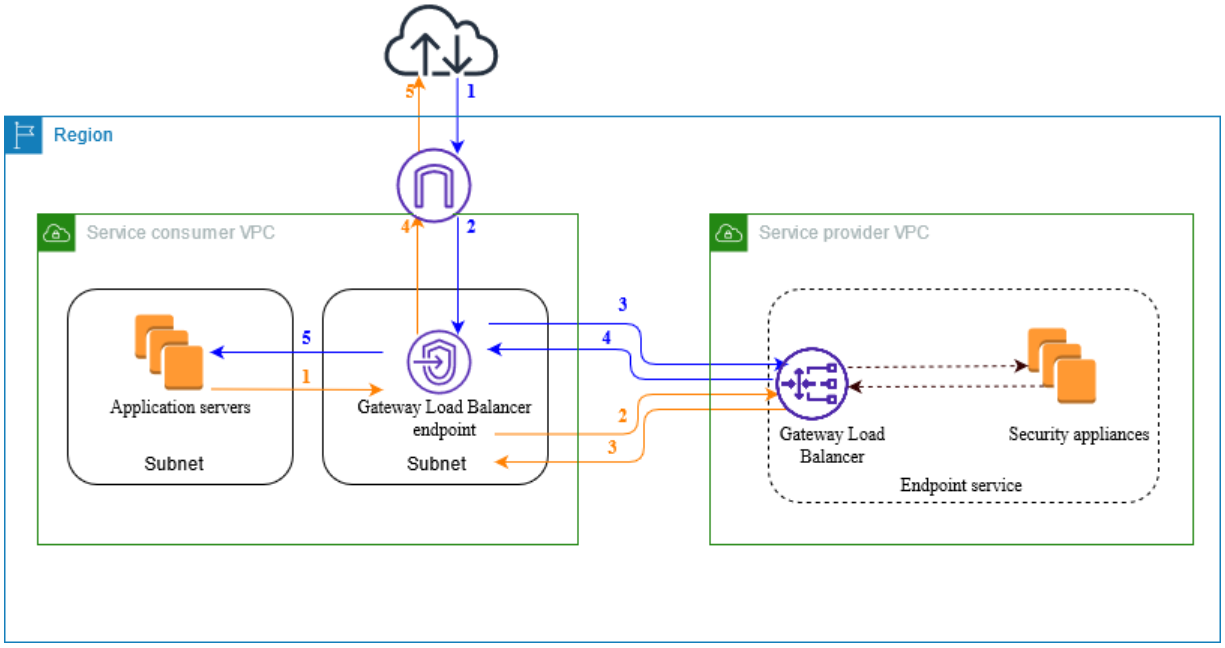
For more information, see Gateway Load Balancers.

Contents

- Overview (p. 38)
- Routing (p. 39)
- Create an inspection system as a Gateway Load Balancer endpoint service (p. 40)
- Access an inspection system using a Gateway Load Balancer endpoint (p. 41)

Overview

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



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

1. Traffic enters the service consumer VPC through the internet gateway.

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

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

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

Routing

To route traffic to the endpoint service, specify the Gateway Load Balancer endpoint as a target in your route tables, using its ID. For the diagram above, add routes to the route tables as follows.

Route table for the internet gateway

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

Destination	Target
vpc-cidr	Local
application-subnet-cidr	*vpc-endpoint-id*

Route table for the subnet with the application servers

This route table must have a route that sends all traffic (0.0.0.0/0) from the application servers to the Gateway Load Balancer endpoint.

Destination	Target
vpc-cidr	Local
0.0.0.0/0	*vpc-endpoint-id*

Route table for the subnet with the Gateway Load Balancer endpoint

This route table must send traffic that is returned from inspection to its final destination. For traffic that originated from the internet, the local route sends the traffic to the application servers. For traffic that originated from the application servers, add a route that sends all traffic (0.0.0.0/0) to the internet gateway.

Destination	Target
vpc-cidr	Local
0.0.0.0/0	*internet-gateway-id*

Create an inspection system as a Gateway Load Balancer endpoint service

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

Endpoint services require either a Network Load Balancer or a Gateway Load Balancer. In this case, you'll create an endpoint service using a Gateway Load Balancer. For more information about creating an endpoint service using a Network Load Balancer, see Create an endpoint service (p. 48).

Contents

- Considerations (p. 40)
- Prerequisites (p. 40)
- Create the endpoint service (p. 40)
- Make your endpoint service available (p. 41)

Considerations

- An endpoint service is available in the Region where you created it.
- An endpoint service supports only IPv4 traffic.
- When service consumers retrieve information about an endpoint service, they can see only the Availability Zones that they have in common with the service provider. When the service provider and service consumer are in different accounts, an Availability Zone name, such as `us-east-1a`, might be mapped to a different physical Availability Zone in each AWS account. You can use AZ IDs to consistently identify the Availability Zones for your service. For more information, see AZ IDs in the *Amazon EC2 User Guide for Linux Instances*.
- There are quotas on your AWS PrivateLink resources. For more information, see AWS PrivateLink quotas (p. 69).

Prerequisites

- Create a service provider VPC with at two subnets in the Availability Zone in which the service should be available. One subnet is for the security appliance instances and the other is for the Gateway Load Balancer.
- Create a Gateway Load Balancer in your service provider VPC. For more information, see Getting started with Gateway Load Balancers.
- Launch security appliances in the service provider VPC and register them with a load balancer target group.

Create the endpoint service

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

To create an endpoint service using the console

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

3. Choose **Create endpoint service**.
4. For **Load balancer type**, choose **Gateway**.
5. For **Available load balancers**, select the Gateway Load Balancer to associate with the endpoint service.
6. For **Require acceptance for endpoint**, select **Acceptance required** to require that connection requests to your endpoint service are accepted manually. Otherwise, these requests are accepted automatically.
7. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
8. Choose **Create**.

To create an endpoint service using the command line

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

Make your endpoint service available

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

- Add permissions that allow each service consumer to connect to your endpoint service. For more information, see the section called “Manage permissions” (p. 51).
- Provide the service consumer with the name of your service and the supported Availability Zones so that they can create an interface endpoint to connect to your service. For more information, see the procedure below.
- Accept the endpoint connection request from the service consumer. For more information see the section called “Accept or reject connection requests” (p. 52).

AWS principals, such as AWS accounts, IAM users, and IAM roles can connect to your endpoint service privately by creating a Gateway Load Balancer endpoint. For more information, see Create a Gateway Load Balancer endpoint (p. 41).

Access an inspection system using a Gateway Load Balancer endpoint

You can create a Gateway Load Balancer endpoint to connect to services powered by AWS PrivateLink.

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

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

Contents

- Considerations (p. 42)
- Prerequisites (p. 42)
- Create the endpoint (p. 42)

- Configure routing (p. 43)
- Manage tags (p. 43)
- Delete a Gateway Load Balancer endpoint (p. 44)

Considerations

- You can choose only one Availability Zone in the service consumer VPC. You can't change this subnet later on. To use a Gateway Load Balancer endpoint in a different subnet, you must create a new Gateway Load Balancer endpoint.
- You can create a single Gateway Load Balancer endpoint per Availability Zone per service, and you must select the Availability Zone that the Gateway Load Balancer supports. When the service provider and service consumer are in different accounts, an Availability Zone name, such as `us-east-1a`, might be mapped to a different physical Availability Zone in each AWS account. You can use AZ IDs to consistently identify the Availability Zones for your service. For more information, see AZ IDs in the *Amazon EC2 User Guide for Linux Instances*.
- Before you can use the endpoint service the service provider must accept the connection requests. The service can't initiate requests to resources in your VPC through the VPC endpoint. The endpoint only returns responses to traffic that was initiated by resources in your VPC.
- Each Gateway Load Balancer endpoint supports a bandwidth of up to 40 Gbps.
- If an endpoint service is associated with multiple Gateway Load Balancers, a Gateway Load Balancer endpoint establishes a connection with only one load balancer per Availability Zone.
- To keep traffic within the same Availability Zone, we recommend that you create a Gateway Load Balancer endpoint in each Availability Zone to which you'll send traffic.
- Network Load Balancer client IP preservation is not supported when traffic is routed through a Gateway Load Balancer endpoint, even if the target is in the same VPC as the Network Load Balancer.
- There are quotas on your AWS PrivateLink resources. For more information, see AWS PrivateLink quotas (p. 69).

Prerequisites

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

Create the endpoint

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

To create a Gateway Load Balancer endpoint using the console

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

4. For **Service category**, choose **Other endpoint services**.
5. For **Service name**, enter the name of the service and choose **Verify service**.
6. For **VPC**, select the VPC in which to create the endpoint.
7. For **Subnets**, select the subnet (Availability Zone) from which you'll access the endpoint service.
8. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
9. Choose **Create endpoint**.

To create a Gateway Load Balancer endpoint using the command line

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

Configure routing

Use the following procedure to configure route tables for the service consumer VPC. This enables the security appliances to perform security inspection for inbound traffic that's destined for the application servers. For more information, see the section called “Routing” (p. 39).

To configure routing using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Route Tables**.
3. Select the route table for the internet gateway and do the following:
   a. Choose **Actions**, **Edit routes**.
   b. Choose **Add route**. For **Destination**, enter the CIDR block of the subnet for the application servers (for example, 10.0.1.0/24). For **Target**, select the VPC endpoint.
   c. Choose **Save routes**.
4. Select the route table for the subnet with the application servers and do the following:
   a. Choose **Actions**, **Edit routes**.
   b. Choose **Add route**. For **Destination**, enter `0.0.0.0/0`. For **Target**, select the VPC endpoint.
   c. Choose **Save routes**.
5. Select the route table for the subnet with the Gateway Load Balancer endpoint, and do the following:
   a. Choose **Actions**, **Edit routes**.
   b. Choose **Add route**. For **Destination**, enter `0.0.0.0/0`. For **Target**, select the internet gateway.
   c. Choose **Save routes**.

To configure routing using the command line

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

Manage tags

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

To manage tags using the console

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

To manage tags using the command line

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

Delete a Gateway Load Balancer endpoint

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

To delete a Gateway Load Balancer endpoint

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

To delete a Gateway Load Balancer endpoint

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

Share your services through AWS PrivateLink

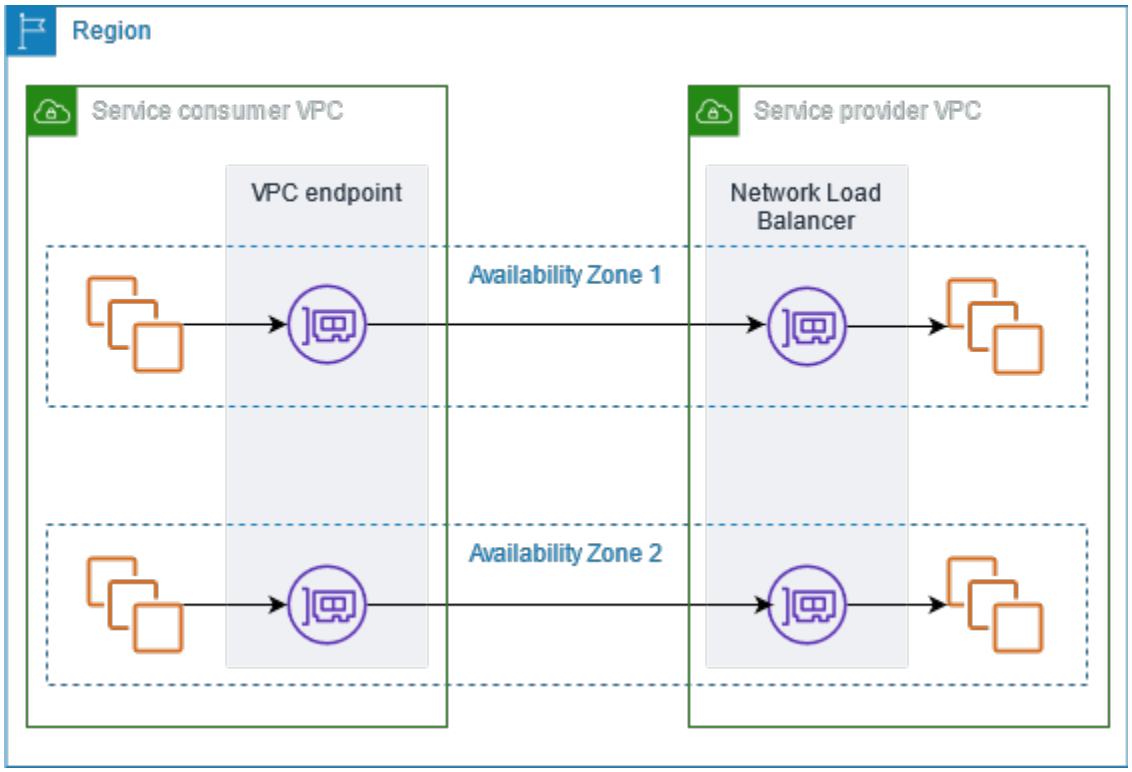
You can host your own AWS PrivateLink powered service, known as an *endpoint service*, and share it with other AWS customers.

Contents

- Overview (p. 45)
- DNS hostnames (p. 46)
- IP address types (p. 47)
- Create a service powered by AWS PrivateLink (p. 48)
- Configure an endpoint service (p. 51)
- Manage DNS names for VPC endpoint services (p. 55)
- Receive alerts for endpoint service events (p. 58)
- Delete an endpoint service (p. 59)

Overview

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



For low latency and fault tolerance, we recommend that you make your service available in all Availability Zones in the Region.

DNS hostnames

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

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

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

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

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

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

Zonal names have the following syntax:

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

A service provider can also associate a private DNS name for their endpoint service, so that service consumers can continue to access the service using its existing DNS name. If the service provider

associated a private DNS name with the endpoint service, the service consumer can enable private DNS names for the interface endpoint. If the service provider doesn't enable private DNS, the service consumer might need to update their application to use the public DNS name for the VPC endpoint service. For more information, see Manage DNS names (p. 55).

IP address types

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

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

Requirements to enable IPv6 for an endpoint service

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

Requirements to enable IPv6 for an interface endpoint

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

DNS record IP address type for an interface endpoint

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

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

Create a service powered by AWS PrivateLink

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

Endpoint services require either a Network Load Balancer or a Gateway Load Balancer. The load balancer receives requests from service consumers and routes them to your service. In this case, you'll create an endpoint service using a Network Load Balancer. For more information about creating an endpoint service using a Gateway Load Balancer, see Access virtual appliances (p. 38).

Contents

- Considerations (p. 48)
- Prerequisites (p. 48)
- Create an endpoint service (p. 49)
- Make your endpoint service available to service consumers (p. 49)

Considerations

- An endpoint service is available in the Region where you created it. You can access the endpoint service from other Regions using VPC peering.
- An endpoint service supports traffic only over TCP.
- When service consumers retrieve information about an endpoint service, they can see only the Availability Zones that they have in common with the service provider. When the service provider and service consumer are in different accounts, an Availability Zone name, such as `us-east-1a`, might be mapped to a different physical Availability Zone in each AWS account. You can use AZ IDs to consistently identify the Availability Zones for your service. For more information, see AZ IDs in the *Amazon EC2 User Guide for Linux Instances*.
- When service consumers send traffic to a service through an interface endpoint, the source IP addresses provided to the application are the private IP addresses of the load balancer nodes, not the IP addresses of the service consumers. If you enable proxy protocol on the load balancer, you can obtain the addresses of the service consumers and the IDs of the interface endpoints from the proxy protocol header. For more information, see Proxy protocol in the *User Guide for Network Load Balancers*.
- If an endpoint service is associated with multiple Network Load Balancers, an interface endpoint establishes a connection with only one load balancer per Availability Zone.
- There are quotas on your AWS PrivateLink resources. For more information, see AWS PrivateLink quotas (p. 69).

Prerequisites

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

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

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

Create an endpoint service

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

To create an endpoint service using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoint Services**.
3. Choose **Create endpoint service**.
4. For **Load balancer type**, choose **Network**.
5. For **Available load balancers**, select the Network Load Balancers to associate with the endpoint service.
6. For **Require acceptance for endpoint**, select **Acceptance required** to require that connection requests to your endpoint service are accepted manually. Otherwise, these requests are accepted automatically.
7. For **Enable private DNS name**, select **Associate a private DNS name with the service** to associate a private DNS name that service consumers can use to access your service, and then enter the private DNS name. Otherwise, service consumers can use the endpoint-specific DNS name provided by AWS. Before service consumers can use the private DNS name, the service provider must verify that they own the domain. For more information, see Manage DNS names (p. 55).
8. For **Supported IP address types**, do one of the following:
   - Select **IPv4** – Enable the endpoint service to accept IPv4 requests.
   - Select **IPv6** – Enable the endpoint service to accept IPv6 requests.
   - Select **IPv4** and **IPv6** – Enable the endpoint service to accept both IPv4 and IPv6 requests.
9. (Optional) To add a tag, choose **Add new tag** and enter the tag key and the tag value.
10. Choose **Create**.

To create an endpoint service using the command line

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

Make your endpoint service available to service consumers

AWS principals, such as AWS accounts, IAM users, and IAM roles can connect to your endpoint service privately by creating an interface VPC endpoint. Service providers must do the following to make their services available to service consumers.

- Add permissions that allow each service consumer to connect to your endpoint service. For more information, see the section called “Manage permissions” (p. 51).
- Provide the service consumer with the name of your service and the supported Availability Zones so that they can create an interface endpoint to connect to your service. For more information, see the following procedure.
- Accept the endpoint connection request from the service consumer. For more information, see the section called “Accept or reject connection requests” (p. 52).

Connect to an endpoint service as the service consumer

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

To create an interface endpoint using the console

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

To create an interface endpoint using the command line

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

Configure an endpoint service

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

Tasks

- Manage permissions (p. 51)
- Accept or reject connection requests (p. 52)
- Change the load balancer association (p. 53)
- Associate a private DNS name (p. 53)
- Modify the supported IP address types (p. 54)
- Manage tags (p. 54)

Manage permissions

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

By default, your endpoint service is not available to service consumers. You must add permissions that allow specific AWS accounts, IAM users, and IAM roles to create an interface VPC endpoint to connect to your endpoint service. To add permissions for an AWS principal, you need its Amazon Resource Name (ARN).

ARNs for AWS principals

AWS account (includes all principals in the account)

arn:aws:iam::*account_id*:root

IAM user

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

IAM role

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

All principals in all AWS accounts

*

Consideration

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

To add permissions for your endpoint service using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Endpoint Services**.
3. Select the endpoint service.
4. Choose **Actions**, **Allow principals**.
5. For **Principals to add**, enter the ARN of the principal. To add another principal, choose **Add principal**.

6. To remove permissions, select the principal and choose **Delete**. When prompted for confirmation, enter **`delete`** and then choose **Delete**.
7. Choose **Allow principals**.

To add permissions for your endpoint service using the command line

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

Accept or reject connection requests

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

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

You can receive a notification when a connection request is accepted or rejected. For more information, see the section called “Receive alerts for endpoint service events” (p. 58).

Consideration

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

To modify the acceptance setting using the console

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

To modify the acceptance setting using the command line

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

To accept or reject a connection request using the console

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

To accept or reject a connection request using the command line

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

Change the load balancer association

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

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

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

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

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

Associate a private DNS name

You can associate a private DNS name with your endpoint service. After you associate a private DNS name, you must update the entry for the domain on your DNS server. Before service consumers can use the private DNS name, the service provider must verify that they own the domain. For more information, see Manage DNS names (p. 55).

To modify an endpoint service private DNS name using the console

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

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

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

To initiate the domain verification process using the console

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

To initiate the domain verification process using the command line

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

Modify the supported IP address types

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

Consideration

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

To modify the supported IP address types using the console

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

To modify the supported IP address types using the command line

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

Manage tags

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

To manage tags using the console

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

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

To add and remove tags using the command line

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

Manage DNS names for VPC endpoint services

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

Considerations

- An endpoint service can have only one private DNS name.
- Private DNS names are not supported for Gateway Load Balancer endpoints.
- To verify a domain, you must have a public hostname or a public DNS provider.
- You can verify the domain of a subdomain. For example, you can verify *example.com*, instead of *a.example.com*. As specified in RFC 1034, each DNS label can have up to 63 characters and the whole domain name must not exceed a total length of 255 characters.

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

Domain ownership verification

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

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

After you add a record, you can check the status of the domain verification process using the Amazon VPC console. In the navigation pane, choose **Endpoint Services**. Select the endpoint service and check the value of **Domain verification status** in the **Details** tab. If domain verification is pending, wait a few minutes and refresh the screen. If needed, you can initiate the verification process manually. Choose **Actions**, **Verify domain ownership for private DNS name**.

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

If the verification status is **failed**, see the section called "Troubleshoot domain verification issues" (p. 58).

Get the name and value

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

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

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

```
[
    {
        "State": "pendingVerification",
        "Type": "TXT",
        "Value": "vpce:l6p0ERxlTt45jevFwOCp",
        "Name": "_6e86v84tqgqubxbwii1m"
    }
]
```

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

Name	Type	Value
_6e86v84tqgqubxbwii1m.example.com	TXT	vpce:l6p0ERxlTt45jevFwOCp

We suggest that you use `Name` as the record subdomain because the base domain name might already be in use. However, if your DNS provider does not allow DNS record names to contain underscores, you can omit the "_6e86v84tqgqubxbwii1m" and simply use "example.com" in the TXT record.

After we verify "_6e86v84tqgqubxbwii1m.example.com", service consumers can use "example.com" or a subdomain (for example, "service.example.com" or "my.service.example.com").

Add a TXT record to your domain's DNS server

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

Amazon Route 53

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

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

For more information, see Create records using the console in the *Amazon Route 53 Developer Guide*.

General procedure

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

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

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

Check whether the TXT record is published

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

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

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

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

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

   The output lists the name servers that serve your domain. You'll query one of these servers in the next step.
2. Verify that the TXT record is correctly published using the following command, where *name_server* is one of the name servers that you found in the previous step.

   ```
   nslookup -type=TXT  _aksldja21i1.example.com name_server
   ```
3. In the output of the previous step, verify that the string that follows `text =` matches the TXT value.

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

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

Troubleshoot domain verification issues

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

- Check whether your DNS provider allows underscores in TXT record names. If this is true for your provider, you can omit the domain verification name (for example, *_aksldja21i1*) from the TXT record.
- Check whether your DNS provider appended the domain name to the end of the TXT record. Some DNS providers automatically append the name of your domain to the attribute name of the TXT record. To avoid this duplication of the domain name, add a period to the end of the domain name when you create the TXT record. This tells your DNS provider that it isn't necessary to append the domain name to the TXT record.
- Check whether your DNS provider modified the DNS record value to use only lowercase letters. We verify your domain only when there is a verification record with an attribute value that exactly matches the value that we provided. If the DNS provider changed your TXT record values to use only lowercase letters, contact them for assistance.
- You might need to verify your domain more than once because you're supporting multiple Regions or multiple AWS accounts. If your DNS provider doesn't allow you to have more than one TXT record with the same attribute name, check whether your DNS provider allows you to assign multiple attribute values to the same TXT record. For example, if your DNS is managed by Amazon Route 53, you can use the following procedure.
  1. In the Route 53 console, choose the TXT record that you created when you verified your domain in the first Region.
  2. For **Value**, go to the end of the existing attribute value, and then press Enter.
  3. Add the attribute value for the additional Region, and then save the record set.

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

Receive alerts for endpoint service events

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

Requirement

Create an Amazon SNS topic for the notifications and subscribe to the topic. Add an access policy to the topic that allows the Amazon VPC endpoint service to publish notifications on your behalf, such as the following. For more information, see How do I edit my Amazon SNS topic's access policy?

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vpce.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "arn:aws:sns:region_code:account_id:topic_name"
```

```
    }
  ]
}
```

To create a notification for an endpoint service using the console

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

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

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

Delete an endpoint service

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

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

To delete an endpoint service using the console

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

To delete an endpoint service using the command line

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

Identity and access management for 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. 62).

Control VPC endpoints creation based on the service owner

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

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

```
            "Condition": {
                "StringEquals": {
                    "ec2:VpceServiceOwner": [
                        "amazon"
                    ]
                }
            }
        }
    ]
}
```

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

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

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ModifyVpcEndpointServiceConfiguration",
                "ec2:CreateVpcEndpointServiceConfiguration"
            ],
            "Resource": [
                "arn:aws:ec2:region:account-id:vpc-endpoint-service/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:VpceServicePrivateDnsName": [
                        "example.com"
                    ]
                }
            }
        }
    ]
}
```

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

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

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

```
            "Effect": "Allow",
            "Action": "ec2:CreateVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:region:account-id:vpc-endpoint/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:VpceServiceName": [
                        "com.amazonaws.region.s3"
                    ]
                }
            }
        }
    ]
}
```

Control access to services using endpoint policies

When you create an interface endpoint or a gateway endpoint, you can attach an endpoint policy. The endpoint policy controls which AWS principals (AWS accounts, IAM users, and IAM roles) can use the VPC endpoint to access the endpoint service.

You cannot attach more than one policy to an endpoint. However, you can modify an endpoint policy at any time.

An endpoint policy does not override or replace IAM user policies or service-specific policies (such as S3 bucket policies). If you're using an interface endpoint to connect to Amazon S3, you can also use Amazon S3 bucket policies to control access to buckets from specific endpoints or specific VPCs.

Contents

- VPC endpoint policies (p. 62)
- Principals for gateway endpoints (p. 63)
- Update a VPC endpoint policy (p. 63)

VPC endpoint policies

A VPC endpoint policy is an IAM resource policy that is attached to an endpoint. If you do not specify an endpoint policy when you create an endpoint, we attach the following policy, which allows full access to the service.

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

Considerations

- Your policy must contain a Principal element.
- The size of an endpoint policy cannot exceed 20,480 characters (including white space).

- Not all endpoint services support endpoint policies. If a service does not support endpoint policies, the endpoint allows full access to the service. For information about the AWS services that support endpoint policies, see the section called “Services that integrate” (p. 9).

For more information about writing policies, see Overview of IAM Policies in the *IAM User Guide*.

Principals for gateway endpoints

With gateway endpoints, if you specify the principal in one of the following formats, access is granted to the account root user only, not all IAM users and roles for the account.

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

```
"AWS": "arn:aws:iam::account_id:root"
```

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

Update a VPC endpoint policy

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

To update a VPC endpoint policy

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

CloudWatch metrics for AWS PrivateLink

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

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

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

For more information, see the Amazon CloudWatch User Guide.

Contents

- Endpoint metrics and dimensions (p. 64)
- Endpoint service metrics and dimensions (p. 66)
- View the CloudWatch metrics (p. 68)

Endpoint metrics and dimensions

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

Metric	Description
`ActiveConnections`	The number of concurrent active connections. This includes connections in the SYN_SENT and ESTABLISHED states. **Reporting criteria**: The endpoint received traffic during the one-minute period. **Statistics**: The most useful statistics are `Average`, `Maximum`, and `Minimum`. **Dimensions** • `Endpoint Type`, `Service Name`, `VPC Endpoint Id`, `VPC Id` • `Endpoint Type`, `Service Name`, `Subnet Id`, `VPC Endpoint Id`, `VPC Id`
`BytesProcessed`	The number of bytes exchanged between endpoints and endpoint services, aggregated in both directions. This is the number of bytes billed to the owner of the endpoint. The bill displays this value in GB.

Metric	Description
	Reporting criteria: The endpoint received traffic during the one-minute period. **Statistics**: The most useful statistics are `Average`, `Sum`, `Maximum`, and `Minimum`. **Dimensions** • `Endpoint Type`, `Service Name`, `VPC Endpoint Id`, `VPC Id` • `Endpoint Type`, `Service Name`, `Subnet Id`, `VPC Endpoint Id`, `VPC Id`
`NewConnections`	The number of new connections established through the endpoint. **Reporting criteria**: The endpoint received traffic during the one-minute period. **Statistics**: The most useful statistics are `Average`, `Sum`, `Maximum`, and `Minimum`. **Dimensions** • `Endpoint Type`, `Service Name`, `VPC Endpoint Id`, `VPC Id` • `Endpoint Type`, `Service Name`, `Subnet Id`, `VPC Endpoint Id`, `VPC Id`
`PacketsDropped`	The number of packets dropped by the endpoint. This metric might not capture all packet drops. Increasing values could indicate that the endpoint or endpoint service is unhealthy. **Reporting criteria**: The endpoint received traffic during the one-minute period. **Statistics**: The most useful statistics are `Average`, `Sum`, and `Maximum`. **Dimensions** • `Endpoint Type`, `Service Name`, `VPC Endpoint Id`, `VPC Id` • `Endpoint Type`, `Service Name`, `Subnet Id`, `VPC Endpoint Id`, `VPC Id`
`RstPacketsReceived`	The number of RST packets received by the endpoint. Increasing values could indicate that the endpoint service is unhealthy. **Reporting criteria**: The endpoint received traffic during the one-minute period. **Statistics**: The most useful statistics are `Average`, `Sum`, and `Maximum`. **Dimensions** • `Endpoint Type`, `Service Name`, `VPC Endpoint Id`, `VPC Id` • `Endpoint Type`, `Service Name`, `Subnet Id`, `VPC Endpoint Id`, `VPC Id`

To filter these metrics, use the following dimensions.

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

Endpoint service metrics and dimensions

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

Metric	Description
`ActiveConnections`	The maximum number of active connections from clients to targets through the endpoints. Increasing values could indicate the need to add targets to the load balancer. **Reporting criteria**: An endpoint connected to the endpoint service sent traffic during the one-minute period. **Statistics**: The most useful statistics are `Average` and `Maximum`. **Dimensions** • `Service Id` • `Az`, `Service Id` • `Load Balancer Arn`, `Service Id` • `Az`, `Load Balancer Arn`, `Service Id` • `Service Id`, `VPC Endpoint Id`
`BytesProcessed`	The number of bytes exchanged between endpoint services and endpoints, in both directions. **Reporting criteria**: An endpoint connected to the endpoint service sent traffic during the one-minute period. **Statistics**: The most useful statistics are `Average`, `Sum`, and `Maximum`. **Dimensions** • `Service Id` • `Az`, `Service Id` • `Load Balancer Arn`, `Service Id` • `Az`, `Load Balancer Arn`, `Service Id` • `Service Id`, `VPC Endpoint Id`
`EndpointsCount`	The number of endpoints connected to the endpoint service.

Metric	Description
	Reporting criteria: There is a nonzero value during the five-minute period. **Statistics**: The most useful statistics are `Average` and `Maximum`. **Dimensions** • `Service Id` • `Az`, `Service Id` • `Load Balancer Arn`, `Service Id` • `Az`, `Load Balancer Arn`, `Service Id` • `Service Id`, `VPC Endpoint Id`
`NewConnections`	The number of new connections established from clients to targets through the endpoints. Increasing values could indicate the need to add targets to the load balancer. **Reporting criteria**: An endpoint connected to the endpoint service sent traffic during the one-minute period. **Statistics**: The most useful statistics are `Average`, `Sum`, and `Maximum`. **Dimensions** • `Service Id` • `Az`, `Service Id` • `Load Balancer Arn`, `Service Id` • `Az`, `Load Balancer Arn`, `Service Id` • `Service Id`, `VPC Endpoint Id`
`RstPacketsSent`	The number of RST packets sent to endpoints by the endpoint service. Increasing values could indicate that there are unhealthy targets. **Reporting criteria**: An endpoint connected to the endpoint service sent traffic during the one-minute period. **Statistics**: The most useful statistics are `Average`, `Sum`, and `Maximum`. **Dimensions** • `Service Id` • `Az`, `Service Id` • `Load Balancer Arn`, `Service Id` • `Az`, `Load Balancer Arn`, `Service Id` • `Service Id`, `VPC Endpoint Id`

To filter these metrics, use the following dimensions.

Dimension	Description
`Az`	Filters the metric data by Availability Zone.
`Load Balancer Arn`	Filters the metric data by load balancer.

Dimension	Description
`Service Id`	Filters the metric data by endpoint service.
`VPC Endpoint Id`	Filters the metric data by VPC endpoint.

View the CloudWatch metrics

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

To view metrics using the Amazon VPC console

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

To view metrics using the CloudWatch console

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

To view metrics using the AWS CLI

Use the following list-metrics command to list the available metrics for interface endpoints and Gateway Load Balancer endpoints:

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

Use the following list-metrics command to list the available metrics for endpoint services:

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

AWS PrivateLink quotas

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

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

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

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

- By default, each interface endpoint can support a bandwidth of up to 10 Gbps per Availability Zone and automatically scales up to 40 Gbps. If your application needs higher throughput, contact AWS support.
- The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed through 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.

Document history for AWS PrivateLink

The following table describes the releases for AWS PrivateLink.

update-history-change	update-history-description	update-history-date
IPv6 support	Service providers can enable their endpoint service to accept IPv6 requests, even if their backend services support only IPv4. If an endpoint service accepts IPv6 requests, service consumers can enable IPv6 support for their interface endpoints so that they can access the endpoint service over IPv6.	May 11, 2022
CloudWatch metrics	AWS PrivateLink publishes CloudWatch metrics for your interface endpoints, Gateway Load Balancer endpoints, and endpoint services.	January 27, 2022
Gateway Load Balancer endpoints	You can create a Gateway Load Balancer endpoint in your VPC to route traffic to a VPC endpoint service that you've configured using a Gateway Load Balancer.	November 10, 2020
VPC endpoint policies	You can attach an IAM policy to an interface VPC endpoint for an AWS service to control access to the service.	March 23, 2020
Condition keys for VPC endpoints and endpoint services	You can use EC2 condition keys to control access to VPC endpoints and endpoint services.	March 6, 2020
Tag VPC endpoints and endpoint services on creation (p. 70)	You can add tags when you create VPC endpoints and endpoint services.	February 5, 2020
Private DNS names	You can access AWS PrivateLink based services from within your VPC using private DNS names.	January 6, 2020
VPC endpoint services	You can create your own endpoints services and enable other AWS accounts and users to connect to your service through an interface VPC endpoint. You	November 28, 2017

	can offer your endpoint services for subscription in the AWS Marketplace.	
Interface VPC endpoints for AWS services	You can create an interface endpoint to connect to AWS services that integrate with AWS PrivateLink without using an internet gateway or NAT device.	November 8, 2017
VPC endpoints for DynamoDB	You can create a gateway VPC endpoint to access Amazon DynamoDB from your VPC without using an internet gateway or NAT device.	August 16, 2017
VPC endpoints for Amazon S3	You can create a gateway VPC endpoint to access Amazon S3 from your VPC without using an internet gateway or NAT device.	May 11, 2015