AWS Client VPN

Administrator Guide

AWS Client VPN: Administrator Guide

Table of Contents

What is AWS Client VPN?

AWS Client VPN is a managed client-based VPN service that enables you to securely access your AWS resources and resources in your on-premises network. With Client VPN, you can access your resources from any location using an OpenVPN-based VPN client.

Contents

Features of Client VPN

Client VPN offers the following features and functionality:

- **Secure connections** — It provides a secure TLS connection from any location using the OpenVPN client.
- **Managed service** — It is an AWS managed service, so it removes the operational burden of deploying and managing a third-party remote access VPN solution.
- **High availability and elasticity** — It automatically scales to the number of users connecting to your AWS resources and on-premises resources.
- **Authentication** — It supports client authentication using Active Directory, federated authentication, and certificate-based authentication.
- **Granular control** — It enables you to implement custom security controls by defining network-based access rules. These rules can be configured at the granularity of Active Directory groups. You can also implement access control using security groups.
- **Ease of use** — It enables you to access your AWS resources and on-premises resources using a single VPN tunnel.
- **Manageability** — It enables you to view connection logs, which provide details on client connection attempts. You can also manage active client connections, with the ability to terminate active client connections.
- **Deep integration** — It integrates with existing AWS services, including AWS Directory Service and Amazon VPC.

Components of Client VPN

The following are the key concepts for Client VPN:

Client VPN endpoint

The Client VPN endpoint is the resource that you create and configure to enable and manage client VPN sessions. It's the termination point for all client VPN sessions.

Target network

A target network is the network that you associate with a Client VPN endpoint. A subnet from a VPC is a target network. Associating a subnet with a Client VPN endpoint enables you to establish VPN sessions. You can associate multiple subnets with a Client VPN endpoint for high availability. All subnets must be from the same VPC. Each subnet must belong to a different Availability Zone.

Route

Each Client VPN endpoint has a route table that describes the available destination network routes. Each route in the route table specifies the path for traffic to specific resources or networks.

Authorization rules

An authorization rule restricts the users who can access a network. For a specified network, you configure the Active Directory or identity provider (IdP) group that is allowed access. Only users belonging to this group can access the specified network. By default, there are no authorization rules and you must configure authorization rules to enable users to access resources and networks.

Client

The end user connecting to the Client VPN endpoint to establish a VPN session. End users need to download an OpenVPN client and use the Client VPN configuration file that you created to establish a VPN session.

Client CIDR range

An IP address range from which to assign client IP addresses. Each connection to the Client VPN endpoint is assigned a unique IP address from the client CIDR range. You choose the client CIDR range, for example, `10.2.0.0/16`.

Client VPN ports

AWS Client VPN supports ports 443 and 1194 for both TCP and UDP. The default is port 443.

Client VPN network interfaces

When you associate a subnet with your Client VPN endpoint, we create Client VPN network interfaces in that subnet. Traffic that's sent to the VPC from the Client VPN endpoint is sent through a Client VPN network interface. Source network address translation (SNAT) is then applied, where the source IP address from the client CIDR range is translated to the Client VPN network interface IP address.

Connection logging

You can enable connection logging for your Client VPN endpoint to log connection events. You can use this information to run forensics, analyze how your Client VPN endpoint is being used, or debug connection issues.

Self-service portal

Client VPN provides a self-service portal as a web page to end users to download the latest version of the AWS VPN Desktop Client and the latest version of the Client VPN endpoint configuration file, which contains the settings required to connect to their endpoint. The Client VPN endpoint administrator can enable or disable the self-service portal for the Client VPN endpoint. Self-service portal is a Global service backed by service stacks in the Asia Pacific (Tokyo), US East (N. Virginia), and Europe (Ireland) Regions, and in AWS GovCloud (US-West).

Working with Client VPN

You can work with Client VPN in any of the following ways:

Amazon VPC console

The Amazon VPC console provides a web-based user interface for Client VPN. If you've signed up for an AWS account, you can sign into the Amazon VPC console and select Client VPN in the navigation pane.

AWS Command Line Interface (CLI)

The AWS CLI provides direct access to the Client VPN public APIs. It is supported on Windows, macOS, and Linux. For more information about getting started with the AWS CLI, see the AWS Command Line Interface User Guide. For more information about the commands for Client VPN, see the AWS CLI Command Reference.

AWS Tools for Windows PowerShell

AWS provides commands for a broad set of AWS offerings for those who script in the PowerShell environment. For more information about getting started with the AWS Tools for Windows PowerShell, see the AWS Tools for Windows PowerShell User Guide. For more information about the cmdlets for Client VPN, see the AWS Tools for Windows PowerShell Cmdlet Reference.

Query API

The Client VPN HTTPS Query API gives you programmatic access to Client VPN and AWS. The HTTPS Query API lets you issue HTTPS requests directly to the service. When you use the HTTPS API, you must include code to digitally sign requests using your credentials. For more information, see the AWS Client VPN actions.

Limitations and rules of Client VPN

Client VPN has the following rules and limitations:

- Client CIDR ranges cannot overlap with the local CIDR of the VPC in which the associated subnet is located, or any routes manually added to the Client VPN endpoint's route table.
- Client CIDR ranges must have a block size of at least /22 and must not be greater than /12.
- A portion of the addresses in the client CIDR range are used to support the availability model of the Client VPN endpoint, and cannot be assigned to clients. Therefore, we recommend that you assign a CIDR block that contains twice the number of IP addresses that are required to enable the maximum number of concurrent connections that you plan to support on the Client VPN endpoint.
- The client CIDR range cannot be changed after you create the Client VPN endpoint.
- The subnets associated with a Client VPN endpoint must be in the same VPC.
- You cannot associate multiple subnets from the same Availability Zone with a Client VPN endpoint.
- A Client VPN endpoint does not support subnet associations in a dedicated tenancy VPC.
- Client VPN supports IPv4 traffic only. See IPv6 Considerations (p. 68) for details regarding IPv6.
- Client VPN is not Federal Information Processing Standards (FIPS) compliant.
- If multi-factor authentication (MFA) is disabled for your Active Directory, a user password cannot be in the following format.

```
SCRV1:<base64_encoded_string>:<base64_encoded_string>
```

- The self-service portal is not available for clients that authenticate using mutual authentication.
- It is not recommended to connect to the Client VPN endpoint by using IP addresses. Since Client VPN is a managed service, you will occasionally see the IP addresses the DNS name resolves to change. In addition, you will see Client VPN network interfaces deleted and recreated in your Cloud Trail logs as well and this is expected behavior. It is recommended to connect to the Client VPN endpoint using the DNS name provided.
- IP forwarding is currently disabled when using the AWS Client VPN Desktop Application. It has been disabled since the launch of the service on December 18, 2018, in order to address an issue reported by NIST. We understand, however, that some customers may need this functionality for their services. While we do not have a specific date at this time, we do plan to safely enable IP forwarding in an upcoming release.

Pricing for Client VPN

You are charged for each endpoint association and each VPN connection on an hourly basis. For more information, see AWS Client VPN pricing.

You are charged for data transfer out from Amazon EC2 to the internet. For more information, see Data Transfer on the Amazon EC2 On-Demand Pricing age.

If you enable connection logging for your Client VPN endpoint, you must create a CloudWatch Logs log group in your account. Charges apply for using log groups. For more information, see Amazon CloudWatch pricing (under **Paid tier**, choose **Logs**).

If you enable the client connect handler for your Client VPN endpoint, you must create and invoke a Lambda function. Charges apply for invoking Lambda functions. For more information, see AWS Lambda pricing.

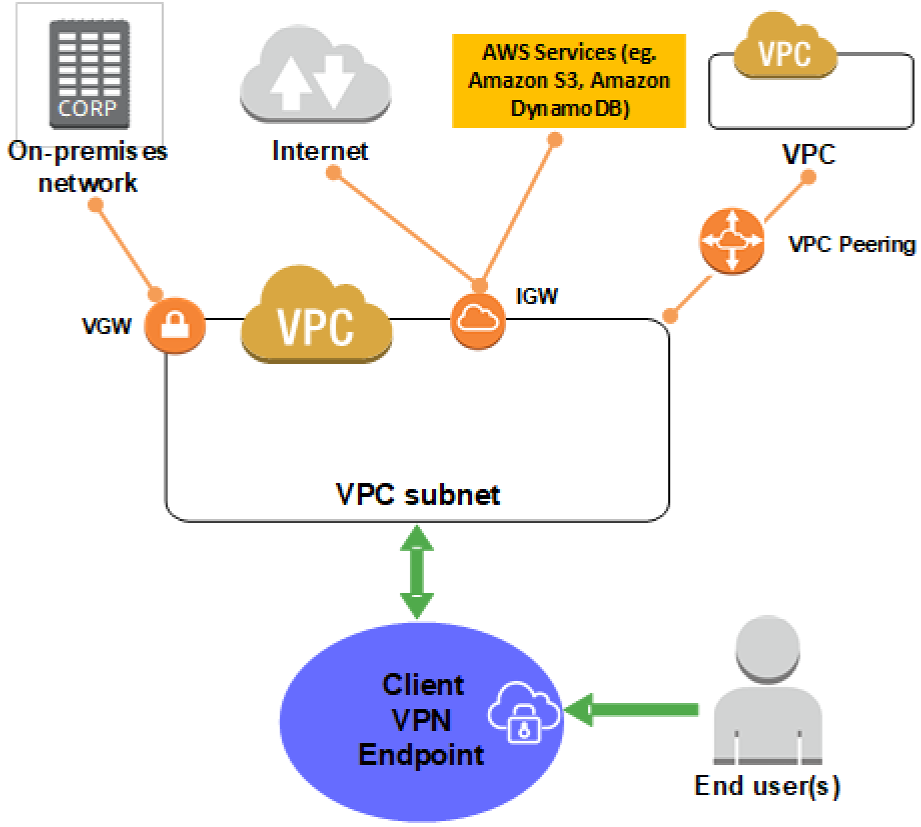
How AWS Client VPN works

With AWS Client VPN, there are two types of user personas that interact with the Client VPN endpoint: administrators and clients.

The *administrator* is responsible for setting up and configuring the service. This involves creating the Client VPN endpoint, associating the target network, and configuring the authorization rules, and setting up additional routes (if required). After the Client VPN endpoint is set up and configured, the administrator downloads the Client VPN endpoint configuration file and distributes it to the clients who need access. The Client VPN endpoint configuration file includes the DNS name of the Client VPN endpoint and authentication information required to establish a VPN session. For more information about setting up the service, see Getting started with Client VPN (p. 35).

The *client* is the end user. This is the person who connects to the Client VPN endpoint to establish a VPN session. The client establishes the VPN session from their local computer or mobile device using an OpenVPN-based VPN client application. After they have established the VPN session, they can securely access the resources in the VPC in which the associated subnet is located. They can also access other resources in AWS, an on-premises network, or other clients if the required route and authorization rules have been configured. For more information about connecting to a Client VPN endpoint to establish a VPN session, see Getting Started in the *AWS Client VPN User Guide*.

The following graphic illustrates the basic Client VPN architecture.



Client authentication

Client authentication is implemented at the first point of entry into the AWS Cloud. It is used to determine whether clients are allowed to connect to the Client VPN endpoint. If authentication succeeds, clients connect to the Client VPN endpoint and establish a VPN session. If authentication fails, the connection is denied and the client is prevented from establishing a VPN session.

Client VPN offers the following types of client authentication:

- Active Directory authentication (p. 6) (user-based)
- Mutual authentication (p. 6) (certificate-based)
- Single sign-on (SAML-based federated authentication) (p. 9) (user-based)

You can use one of methods listed above alone, or a combination of mutual authentication with a user-based method such as the following:

- Mutual authentication and federated authentication
- Mutual authentication and Active Directory authentication

> **Important**
> To create a Client VPN endpoint, you must provision a server certificate in AWS Certificate Manager, regardless of the type of authentication you use. For more information about creating and provisioning a server certificate, see the steps in Mutual authentication (p. 6).

Active Directory authentication

Client VPN provides Active Directory support by integrating with AWS Directory Service. With Active Directory authentication, clients are authenticated against existing Active Directory groups. Using AWS Directory Service, Client VPN can connect to existing Active Directories provisioned in AWS or in your on-premises network. This allows you to use your existing client authentication infrastructure. If you are using an on-premises Active Directory and you do not have an existing AWS Managed Microsoft AD, you must configure an Active Directory Connector (AD Connector). You can use one Active Directory server to authenticate the users. For more information about Active Directory integration, see the AWS Directory Service Administration Guide.

Client VPN supports multi-factor authentication (MFA) when it's enabled for AWS Managed Microsoft AD or AD Connector. If MFA is enabled, clients must enter a user name, password, and MFA code when they connect to a Client VPN endpoint. For more information about enabling MFA, see Enable Multi-Factor Authentication for AWS Managed Microsoft AD and Enable Multi-Factor Authentication for AD Connector in the *AWS Directory Service Administration Guide*.

For quotas and rules for configuring users and groups in Active Directory, see Users and groups quotas (p. 74).

Mutual authentication

With mutual authentication, Client VPN uses certificates to perform authentication between the client and the server. Certificates are a digital form of identification issued by a certificate authority (CA). The server uses client certificates to authenticate clients when they attempt to connect to the Client VPN endpoint. You must create a server certificate and key, and at least one client certificate and key.

You must upload the server certificate to AWS Certificate Manager (ACM) and specify it when you create a Client VPN endpoint. When you upload the server certificate to ACM, you also specify the certificate authority (CA). You only need to upload the client certificate to ACM when the CA of the client certificate is different from the CA of the server certificate. For more information about ACM, see the AWS Certificate Manager User Guide.

You can create a separate client certificate and key for each client that will connect to the Client VPN endpoint. This enables you to revoke a specific client certificate if a user leaves your organization. In this case, when you create the Client VPN endpoint, you can specify the server certificate ARN for the client certificate, provided that the client certificate has been issued by the same CA as the server certificate.

> **Note**
> A Client VPN endpoint supports 1024-bit and 2048-bit RSA key sizes only. Also, the client certificate must have the CN attribute in the Subject field.

Linux/macOS

The following procedure uses OpenVPN easy-rsa to generate the server and client certificates and keys, and then uploads the server certificate and key to ACM. For more information, see the Easy-RSA 3 Quickstart README.

To generate the server and client certificates and keys and upload them to ACM

1. Clone the OpenVPN easy-rsa repo to your local computer and navigate to the `easy-rsa/easyrsa3` folder.

```
$ git clone https://github.com/OpenVPN/easy-rsa.git
```

```
$ cd easy-rsa/easyrsa3
```

2. Initialize a new PKI environment.

```
$ ./easyrsa init-pki
```

3. To build a new certificate authority (CA), run this command and follow the prompts.

```
$ ./easyrsa build-ca nopass
```

4. Generate the server certificate and key.

```
$ ./easyrsa build-server-full server nopass
```

5. Generate the client certificate and key.

   Make sure to save the client certificate and the client private key because you will need them when you configure the client.

```
$ ./easyrsa build-client-full client1.domain.tld nopass
```

   You can optionally repeat this step for each client (end user) that requires a client certificate and key.

6. Copy the server certificate and key and the client certificate and key to a custom folder and then navigate into the custom folder.

   Before you copy the certificates and keys, create the custom folder by using the `mkdir` command. The following example creates a custom folder in your home directory.

```
$ mkdir ~/custom_folder/
$ cp pki/ca.crt ~/custom_folder/
$ cp pki/issued/server.crt ~/custom_folder/
$ cp pki/private/server.key ~/custom_folder/
$ cp pki/issued/client1.domain.tld.crt ~/custom_folder
$ cp pki/private/client1.domain.tld.key ~/custom_folder/
$ cd ~/custom_folder/
```

7. Upload the server certificate and key and the client certificate and key to ACM. Be sure to upload them in the same Region in which you intend to create the Client VPN endpoint. The following commands use the AWS CLI to upload the certificates. To upload the certificates using the ACM console instead, see Import a certificate in the *AWS Certificate Manager User Guide*.

   ```
   $ aws acm import-certificate --certificate fileb://server.crt --private-key
    fileb://server.key --certificate-chain fileb://ca.crt
   ```

   ```
   $ aws acm import-certificate --certificate fileb://client1.domain.tld.crt --
   private-key fileb://client1.domain.tld.key --certificate-chain fileb://ca.crt
   ```

   You do not necessarily need to upload the client certificate to ACM. If the server and client certificates have been issued by the same Certificate Authority (CA), you can use the server certificate ARN for both server and client when you create the Client VPN endpoint. In the steps above, the same CA has been used to create both certificates. However, the steps to upload the client certificate are included for completeness.

Windows

The following procedure installs Easy-RSA 3.x software and uses it to generate server and client certificates and keys.

To generate server and client certificates and keys and upload them to ACM

1. Open the EasyRSA releases page and download the ZIP file for your version of Windows and extract it.
2. Open a command prompt and navigate to the location that the `EasyRSA-3.x` folder was extracted to.
3. Run the following command to open the EasyRSA 3 shell.

   ```
   C:\Program Files\EasyRSA-3.x> .\EasyRSA-Start.bat
   ```

4. Initialize a new PKI environment.

   ```
   # ./easyrsa init-pki
   ```

5. To build a new certificate authority (CA), run this command and follow the prompts.

   ```
   # ./easyrsa build-ca nopass
   ```

6. Generate the server certificate and key.

   ```
   # ./easyrsa build-server-full server nopass
   ```

7. Generate the client certificate and key.

   ```
   # ./easyrsa build-client-full client1.domain.tld nopass
   ```

   You can optionally repeat this step for each client (end user) that requires a client certificate and key.

8. Exit the EasyRSA 3 shell.

```
# exit
```

9. Copy the server certificate and key and the client certificate and key to a custom folder and then navigate into the custom folder.

   Before you copy the certificates and keys, create the custom folder by using the `mkdir` command. The following example creates a custom folder in your C:\ drive.

```
C:\Program Files\EasyRSA-3.x> mkdir C:\custom_folder
C:\Program Files\EasyRSA-3.x> copy pki\ca.crt C:\custom_folder
C:\Program Files\EasyRSA-3.x> copy pki\issued\server.crt C:\custom_folder
C:\Program Files\EasyRSA-3.x> copy pki\private\server.key C:\custom_folder
C:\Program Files\EasyRSA-3.x> copy pki\issued\client1.domain.tld.crt C:
\custom_folder
C:\Program Files\EasyRSA-3.x> copy pki\private\client1.domain.tld.key C:
\custom_folder
C:\Program Files\EasyRSA-3.x> cd C:\custom_folder
```

10. Upload the server certificate and key and the client certificate and key to ACM. Be sure to upload them in the same Region in which you intend to create the Client VPN endpoint. The following commands use the AWS CLI to upload the certificates. To upload the certificates using the ACM console instead, see Import a certificate in the *AWS Certificate Manager User Guide*.

```
aws acm import-certificate --certificate fileb://server.crt --private-key fileb://
server.key --certificate-chain fileb://ca.crt
```

```
aws acm import-certificate --certificate fileb://client1.domain.tld.crt --private-
key fileb://client1.domain.tld.key --certificate-chain fileb://ca.crt
```

   You do not necessarily need to upload the client certificate to ACM. If the server and client certificates have been issued by the same Certificate Authority (CA), you can use the server certificate ARN for both server and client when you create the Client VPN endpoint. In the steps above, the same CA has been used to create both certificates. However, the steps to upload the client certificate are included for completeness.

Single sign-on (SAML 2.0-based federated authentication)

AWS Client VPN supports identity federation with Security Assertion Markup Language 2.0 (SAML 2.0) for Client VPN endpoints. You can use identity providers (IdPs) that support SAML 2.0 to create centralized user identities. You can then configure a Client VPN endpoint to use SAML-based federated authentication, and associate it with the IdP. Users then connect to the Client VPN endpoint using their centralized credentials.

To enable your SAML-based IdP to work with a Client VPN endpoint, you must do the following.

1. Create a SAML-based app in your chosen IdP to use with AWS Client VPN, or use an existing app.
2. Configure your IdP to establish a trust relationship with AWS. For resources, see SAML-based IdP configuration resources (p. 12).
3. In your IdP, generate and download a federation metadata document that describes your organization as an IdP. This signed XML document is used to establish the trust relationship between AWS and the IdP.

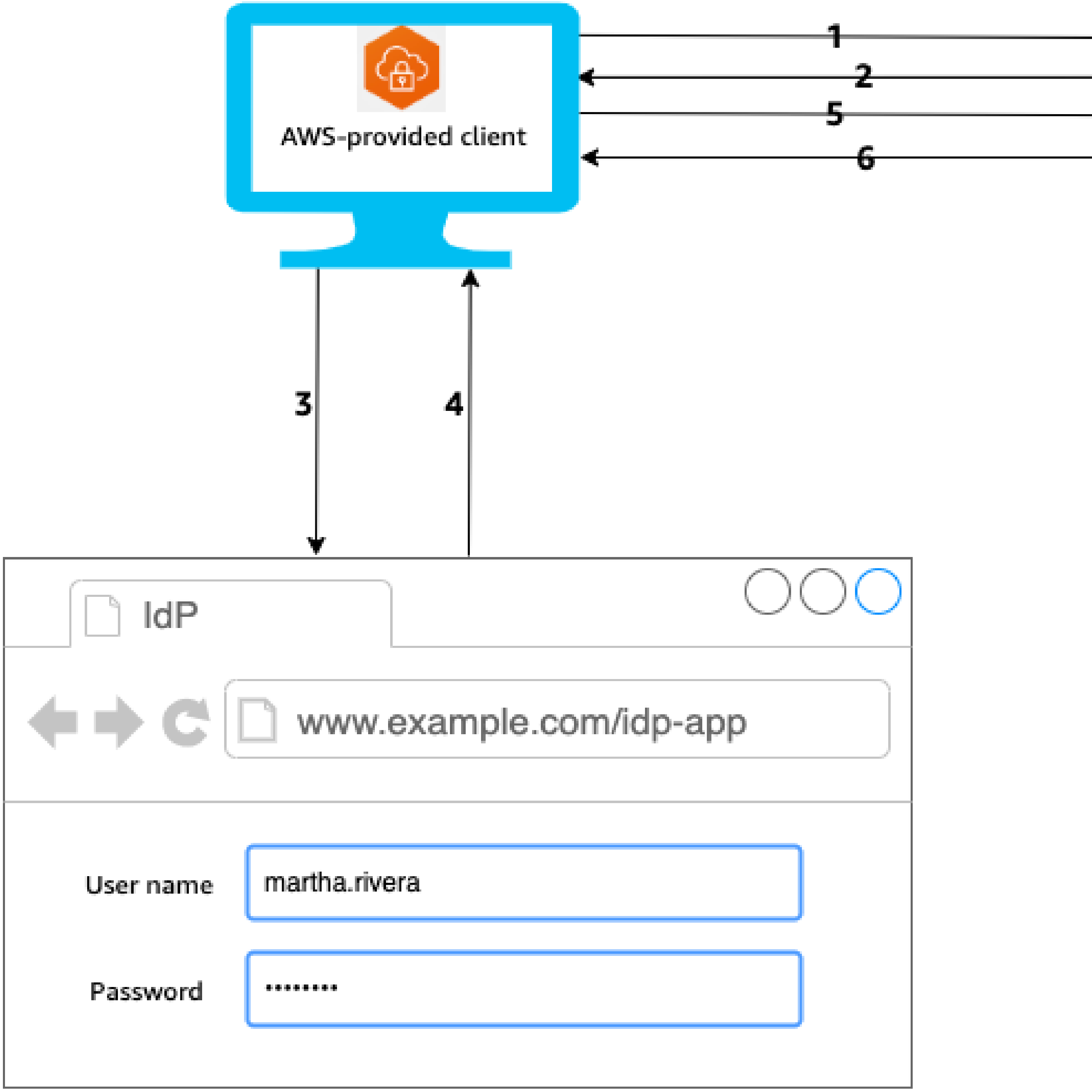
4. Create an IAM SAML identity provider in the same AWS account as the Client VPN endpoint. The IAM SAML identity provider defines your organization's IdP-to-AWS trust relationship using the metadata document generated by the IdP. For more information, see Creating IAM SAML Identity Providers in the *IAM User Guide*. If you later update the app configuration in the IdP, generate a new metadata document and update your IAM SAML identity provider.

   Note
   You do not need to create an IAM role to use the IAM SAML identity provider.

5. Create a Client VPN endpoint. Specify federated authentication as the authentication type, and specify the IAM SAML identity provider that you created. For more information, see Create a Client VPN endpoint (p. 49).
6. Export the client configuration file (p. 55) and distribute it to your users. Instruct your users to download the latest version of the AWS provided client, and to use it to load the configuration file and connect to the Client VPN endpoint. Alternatively, if you enabled the self-service portal for your Client VPN endpoint, instruct your users to go to the self-service portal to get the configuration file and AWS provided client. For more information, see Access the self-service portal (p. 41).

Authentication workflow

The following diagram provides an overview of the authentication workflow for a Client VPN endpoint that uses SAML-based federated authentication. When you create and configure the Client VPN endpoint, you specify the IAM SAML identity provider.



1. The user opens the AWS provided client on their device and initiates a connection to the Client VPN endpoint.
2. The Client VPN endpoint sends an IdP URL and authentication request back to the client, based on the information that was provided in the IAM SAML identity provider.
3. The AWS provided client opens a new browser window on the user's device. The browser makes a request to the IdP and displays a login page.
4. The user enters their credentials on the login page, and the IdP sends a signed SAML assertion back to the client.
5. The AWS provided client sends the SAML assertion to the Client VPN endpoint.
6. The Client VPN endpoint validates the assertion and either allows or denies access to the user.

Requirements and considerations for SAML-based federated authentication

The following are the requirements and considerations for SAML-based federated authentication.

- For quotas and rules for configuring users and groups in a SAML-based IdP, see Users and groups quotas (p. 74).
- The SAML response must be signed and unencrypted.
- The maximum supported size for SAML responses is 128 KB.
- AWS Client VPN does not provide signed authentication requests.
- SAML single logout is not supported. Users can log out by disconnecting from the AWS provided client, or you can terminate the connections (p. 46).
- A Client VPN endpoint supports a single IdP only.
- Multi-factor authentication (MFA) is supported when it's enabled in your IdP.
- Users must use the AWS provided client to connect to the Client VPN endpoint. They must use version 1.2.0 or later. For more information, see Connect using the AWS provided client.
- The following browsers are supported for IdP authentication: Apple Safari, Google Chrome, Microsoft Edge, and Mozilla Firefox.
- The AWS provided client reserves TCP port 35001 on users' devices for the SAML response.
- If the metadata document for the IAM SAML identity provider is updated with an incorrect or malicious URL, this can cause authentication issues for users, or result in phishing attacks. Therefore, we recommend that you use AWS CloudTrail to monitor updates that are made to the IAM SAML identity provider. For more information, see Logging IAM and AWS STS calls with AWS CloudTrail in the *IAM User Guide*.
- AWS Client VPN sends an AuthN request to the IdP via an HTTP Redirect binding. Therefore, the IdP should support HTTP Redirect binding and it should be present in the IdP's metadata document.
- For the SAML assertion, you must use an email address format for the `NameID` attribute.

SAML-based IdP configuration resources

The following table lists the SAML-based IdPs that we have tested for use with AWS Client VPN, and resources that can help you configure the IdP.

IdP	Resource
Okta	Authenticate AWS Client VPN users with SAML
Microsoft Azure Active Directory	For more information, see Tutorial: Azure Active Directory single sign-on (SSO) integration with AWS ClientVPN on the Microsoft documentation website.

Service provider information for creating an app

To create a SAML-based app using an IdP that's not listed in the preceding table, use the following information to configure the AWS Client VPN service provider information.

- Assertion Consumer Service (ACS) URL: `http://127.0.0.1:35001`
- Audience URI: `urn:amazon:webservices:clientvpn`

The following attribute is required.

Attribute	Description
`memberOf`	The group or groups that the user belongs to.

Attributes are case-sensitive, and must be configured exactly as specified.

Support for the self-service portal

If you enable the self-service portal for your Client VPN endpoint, users log into the portal using their SAML-based IdP credentials.

If your IdP supports multiple Assertion Consumer Service (ACS) URLs, add the following ACS URL to your app.

```
https://self-service.clientvpn.amazonaws.com/api/auth/sso/saml
```

If you are using the Client VPN endpoint in a GovCloud region, use the following ACS URL instead. If you use the same IDP app to authenticate for both standard and GovCloud regions, you can add both URLs.

```
https://gov.self-service.clientvpn.amazonaws.com/api/auth/sso/saml
```

If your IdP does not support multiple ACS URLs, do the following:

1. Create an additional SAML-based app in your IdP and specify the following ACS URL.

   ```
   https://self-service.clientvpn.amazonaws.com/api/auth/sso/saml
   ```
2. Generate and download a federation metadata document.
3. Create an IAM SAML identity provider in the same AWS account as the Client VPN endpoint. For more information, see Creating IAM SAML Identity Providers in the *IAM User Guide*.

   Note
   You create this IAM SAML identity provider in addition to the one you create for the main app (p. 9).
4. Create the Client VPN endpoint (p. 49), and specify both of the IAM SAML identity providers that you created.

Client authorization

Client VPN supports two types of client authorization: security groups and network-based authorization (using authorization rules).

Security groups

When you create a Client VPN endpoint, you can specify the security groups from a specific VPC to apply to the Client VPN endpoint. When you associate a subnet with a Client VPN endpoint, we automatically apply the VPC's default security group. You can change the security groups after you create the Client VPN endpoint. For more information, see Apply a security group to a target network (p. 60). The security groups are associated with the Client VPN network interfaces.

You can enable Client VPN users to access your applications in a VPC by adding a rule to your applications' security groups to allow traffic from the security group that was applied to the association.

To add a rule that allows traffic from the Client VPN endpoint security group

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Security Groups**.
3. Choose the security group that's associated with your resource or application, and choose **Actions**, **Edit inbound rules**.
4. Choose **Add rule**.
5. For **Type**, choose **All traffic**. Alternatively, you can restrict access to a specific type of traffic, for example, **SSH**.

   For **Source**, specify the ID of the security group that's associated with the target network (subnet) for the Client VPN endpoint.
6. Choose **Save rules**.

Conversely, you can restrict access for Client VPN users by not specifying the security group that was applied to the association, or by removing the rule that references the Client VPN endpoint security group. The security group rules that you require might also depend on the kind of VPN access that you want to configure. For more information, see Scenarios and examples (p. 22).

For more information about security groups, see Security groups for your VPC in the *Amazon VPC User Guide*.

Network-based authorization

Network-based authorization is implemented using authorization rules. For each network that you want to enable access, you must configure authorization rules that limit the users who have access. For a specified network, you configure the Active Directory group or the SAML-based IdP group that is allowed access. Only users who belong to the specified group can access the specified network. If you are not using Active Directory or SAML-based federated authentication, or you want to open access to all users, you can specify a rule that grants access to all clients. For more information, see Authorization rules (p. 42).

Connection authorization

You can configure a *client connect handler* for your Client VPN endpoint. The handler enables you to run custom logic that authorizes a new connection, based on device, user, and connection attributes. The client connect handler runs after the Client VPN service has authenticated the device and user.

To configure a client connect handler for your Client VPN endpoint, create an AWS Lambda function that takes device, user, and connection attributes as inputs, and returns a decision to the Client VPN service to allow or deny a new connection. You specify the Lambda function in your Client VPN endpoint. When devices connect to your Client VPN endpoint, the Client VPN service invokes the Lambda function on your behalf. Only connections that are authorized by the Lambda function are allowed to connect to the Client VPN endpoint.

> **Note**
> Currently, the only type of client connect handler that is supported is a Lambda function.

Requirements and considerations

The following are the requirements and considerations for the client connect handler:

- The name of the Lambda function must begin with the `AWSClientVPN-` prefix.
- Qualified Lambda functions are supported.
- The Lambda function must be in the same AWS Region and the same AWS account as the Client VPN endpoint.
- The Lambda function times out after 30 seconds. This value cannot be changed.
- The Lambda function is invoked synchronously. It's invoked after device and user authentication, and before the authorization rules are evaluated.
- If the Lambda function is invoked for a new connection and the Client VPN service does not get an expected response from the function, the Client VPN service denies the connection request. For example, this can occur if the Lambda function is throttled, times out, or encounters other unexpected errors, or if the function's response is not in a valid format.
- We recommend that you configure provisioned concurrency for the Lambda function to enable it to scale without fluctuations in latency.
- If you update your Lambda function, existing connections to the Client VPN endpoint are not affected. You can terminate the existing connections, and then instruct your clients to establish new connections. For more information, see Terminate a client connection (p. 46).
- If clients use the AWS provided client to connect to the Client VPN endpoint, they must use version 1.2.6 or later for Windows, and version 1.2.4 or later for macOS. For more information, see Connect using the AWS provided client.

Lambda interface

The Lambda function takes device attributes, user attributes, and connection attributes as inputs from the Client VPN service. It must then return a decision to the Client VPN service whether to allow or deny the connection.

Request schema

The Lambda function takes a JSON blob containing the following fields as input.

```
{
    "connection-id": <connection ID>,
    "endpoint-id": <client VPN endpoint ID>,
    "common-name": <cert-common-name>,
    "username": <user identifier>,
    "platform": <OS platform>,
    "platform-version": <OS version>,
    "public-ip": <public IP address>,
    "client-openvpn-version": <client OpenVPN version>,
    "groups": <group identifier>,
    "schema-version": "v2"
}
```

- `connection-id` — The ID of the client connection to the Client VPN endpoint.
- `endpoint-id` — The ID of the Client VPN endpoint.
- `common-name` — The device identifier. In the client certificate that you create for the device, the common name uniquely identifies the device.
- `username` — The user identifier, if applicable. For Active Directory authentication, this is the user name. For SAML-based federated authentication, this is `NameID`. For mutual authentication, this field is empty.
- `platform` — The client operating system platform.
- `platform-version` — The version of the operating system. The Client VPN service provides a value when the `--push-peer-info` directive is present in the OpenVPN client configuration when clients connect to a Client VPN endpoint, and when the client is running the Windows platform.

- `public-ip` — The public IP address of the connecting device.
- `client-openvpn-version` — The OpenVPN version that the client is using.
- `groups` — The group identifier, if applicable. For Active Directory authentication, this will be a list of Active Directory groups. For SAML-based federated authentication, this will be a list of identity provider (IdP) groups. For mutual authentication, this field is empty.
- `schema-version` — The schema version. The default is `v2`.

Response schema

The Lambda function must return the following fields.

```
{
    "allow": boolean,
    "error-msg-on-denied-connection": "",
    "posture-compliance-statuses": [],
    "schema-version": "v2"
}
```

- `allow` — Required. A boolean (`true` | `false`) that indicates whether to allow or deny the new connection.
- `error-msg-on-denied-connection` — Required. A string of up to 255 characters that can be used to provide steps and guidance to clients if the connection is denied by the Lambda function. In the event of failures during the running of the Lambda function (for example, due to throttling) the following default message is returned to clients.

  ```
  Error establishing connection. Please contact your administrator.
  ```
- `posture-compliance-statuses` — Required. If you use the Lambda function for posture assessment (p. 16), this is a list of statuses for the connecting device. You define the status names according to your posture assessment categories for devices, for example, `compliant`, `quarantined`, `unknown`, and so on. Each name can be up to 255 characters in length. You can specify up to 10 statuses.
- `schema-version` — Required. The schema version. The default is `v2`.

You can use the same Lambda function for multiple Client VPN endpoints in the same Region.

For more information about creating a Lambda function, see Getting started with AWS Lambda in the *AWS Lambda Developer Guide*.

Using the client connect handler for posture assessment

You can use the client connect handler to integrate your Client VPN endpoint with your existing device management solution to evaluate the posture compliance of connecting devices. For the Lambda function to work as a device authorization handler, use mutual authentication (p. 6) for your Client VPN endpoint. Create a unique client certificate and key for each client (device) that will connect to the Client VPN endpoint. The Lambda function can use the unique common name for the client certificate (that's passed from the Client VPN service) to identify the device and fetch its posture compliance status from your device management solution. You can use mutual authentication combined with user-based authentication.

Alternatively, you can do a basic posture assessment in the Lambda function itself. For example, you can assess the `platform` and `platform-version` fields that are passed to the Lambda function by the Client VPN service.

Enabling the client connect handler

To enable the client connect handler, create or modify a Client VPN endpoint and specify the Amazon Resource Name (ARN) of the Lambda function. For more information, see Create a Client VPN endpoint (p. 49) and Modify a Client VPN endpoint (p. 51).

Service-linked role

AWS Client VPN automatically creates a service-linked role in your account called **AWSServiceRoleForClientVPNConnections**. The role has permissions to invoke the Lambda function when a connection is made to the Client VPN endpoint. For more information, see Using service-linked roles for Client VPN (p. 65).

Monitoring connection authorization failures

You can view the connection authorization status of connections to the Client VPN endpoint. For more information, see View client connections (p. 46).

When the client connect handler is used for posture assessment, you can also view the posture compliance statuses of devices that connect to your Client VPN endpoint in the connection logs. For more information, see Connection logging (p. 19).

If a device fails connection authorization, the `connection-attempt-failure-reason` field in the connection logs returns one of the following failure reasons:

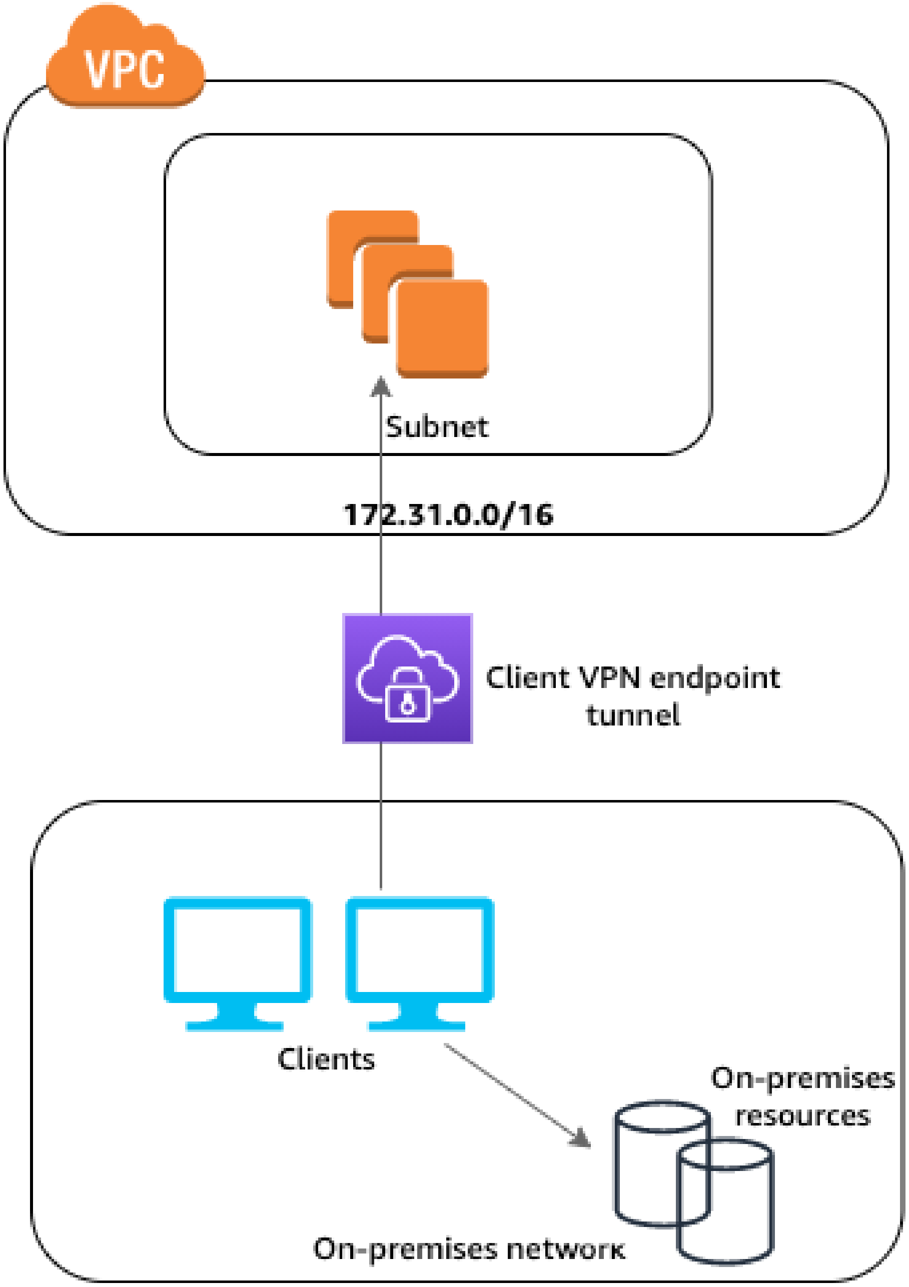
- `client-connect-failed` — The Lambda function prevented the connection from being established.
- `client-connect-handler-timed-out` — The Lambda function timed out.
- `client-connect-handler-other-execution-error` — The Lambda function encountered an unexpected error.
- `client-connect-handler-throttled` — The Lambda function was throttled.
- `client-connect-handler-invalid-response` — The Lambda function returned a response that was not valid.
- `client-connect-handler-service-error` — There was a service-side error during the connection attempt.

Split-tunnel on AWS Client VPN endpoints

By default, when you have a Client VPN endpoint, all traffic from clients is routed over the Client VPN tunnel. When you enable split-tunnel on the Client VPN endpoint, we push the routes on the Client VPN endpoint route table (p. 57) to the device that is connected to the Client VPN endpoint. This ensures that only traffic with a destination to the network matching a route from the Client VPN endpoint route table is routed over the Client VPN tunnel.

You can use a split-tunnel Client VPN endpoint when you do not want all user traffic to route through the Client VPN endpoint.

In the following example, split-tunnel is enabled on the Client VPN endpoint. Only traffic that's destined for the VPC (`172.31.0.0/16`) is routed over the Client VPN tunnel. Traffic that's destined for on-premises resources is not routed over the Client VPN tunnel.



Split-tunnel benefits

Split-tunnel on Client VPN endpoints offers the following benefits:

- You can optimize the routing of traffic from clients by having only the AWS destined traffic traverse the VPN tunnel.
- You can reduce the volume of outgoing traffic from AWS, therefore reducing the data transfer cost.

Routing considerations

When you enable split-tunnel on a Client VPN endpoint, all of the routes that are in the Client VPN route tables are added to the client route table when the VPN is established. This operation is different from the default Client VPN endpoint operation, which overwrites the client route table with the entry 0.0.0.0/0 to route all traffic over the VPN.

Enabling-split-tunnel

You can enable split-tunnel on a new or existing Client VPN endpoint. For more information, see the following topics:

- Create a Client VPN endpoint (p. 49)
- Modify a Client VPN endpoint (p. 51)

Connection logging

Connection logging is a feature of AWS Client VPN that enables you to capture *connection logs* for your Client VPN endpoint.

A connection log contains *connection log entries*. Each connection log entry contains information about a connection event, which is when a client (end user) connects, attempts to connect, or disconnects from your Client VPN endpoint. You can use this information to run forensics, analyze how your Client VPN endpoint is being used, or debug connection issues.

Connection logging is available in all Regions where AWS Client VPN is available. Connection logs are published to a CloudWatch Logs log group in your account.

Connection log entries

A connection log entry is a JSON-formatted blob of key-value pairs. The following is an example connection log entry.

```
{
    "connection-log-type": "connection-attempt",
    "connection-attempt-status": "successful",
    "connection-reset-status": "NA",
    "connection-attempt-failure-reason": "NA",
    "connection-id": "cvpn-connection-abc123abc123abc12",
    "client-vpn-endpoint-id": "cvpn-endpoint-aaa111bbb222ccc33",
    "transport-protocol": "udp",
    "connection-start-time": "2020-03-26 20:37:15",
    "connection-last-update-time": "2020-03-26 20:37:15",
    "client-ip": "10.0.1.2",
    "common-name": "client1",
    "device-type": "mac",
    "device-ip": "98.247.202.82",
    "port": "50096",
    "ingress-bytes": "0",
    "egress-bytes": "0",
    "ingress-packets": "0",
    "egress-packets": "0",
    "connection-end-time": "NA"
}
```

A connection log entry contains the following keys:

- `connection-log-type` — The type of connection log entry (`connection-attempt` or `connection-reset`).
- `connection-attempt-status` — The status of the connection request (`successful`, `failed`, `waiting-for-assertion`, or `NA`).
- `connection-reset-status` — The status of a connection reset event (`NA` or `assertion-received`).
- `connection-attempt-failure-reason` — The reason for the connection failure, if applicable.
- `connection-id` — The ID of the connection.
- `client-vpn-endpoint-id` — The ID of the Client VPN endpoint to which the connection was made.
- `transport-protocol` — The transport protocol that was used for the connection.
- `connection-start-time` — The start time of the connection.
- `connection-last-update-time` — The last update time of the connection. This value is periodically updated in the logs.
- `client-ip` — The IP address of the client, which is allocated from the client IPv4 CIDR range for the Client VPN endpoint.
- `common-name` — The common name of the certificate that's used for certificate-based authentication.
- `device-type` — The type of device used for the connection by the end user.
- `device-ip` — The public IP address of the device.
- `port` — The port number for the connection.
- `ingress-bytes` — The number of ingress (inbound) bytes for the connection. This value is periodically updated in the logs.
- `egress-bytes` — The number of egress (outbound) bytes for the connection. This value is periodically updated in the logs.
- `ingress-packets` — The number of ingress (inbound) packets for the connection. This value is periodically updated in the logs.
- `egress-packets` — The number of egress (outbound) packets for the connection. This value is periodically updated in the logs.
- `connection-end-time` — The end time of the connection. The value is `NA` if the connection is still in progress or if the connection attempt failed.
- `posture-compliance-statuses` — The posture compliance statuses returned by the client connect handler (p. 14), if applicable.

For more information about enabling connection logging, see Working with connection logs (p. 53).

Client VPN scaling considerations

When you create a Client VPN endpoint, consider the maximum number of concurrent VPN connections that you plan to support. You should take into account the number of clients that you currently support, and whether your Client VPN endpoint can meet additional demand if needed.

The following factors affect the maximum number of concurrent VPN connections that can be supported on a Client VPN endpoint.

Client CIDR range size

When you create a Client VPN endpoint (p. 49), you must specify a client CIDR range, which is an IPv4 CIDR block between a /12 and /22 netmask. Each VPN connection to the Client VPN endpoint is assigned a unique IP address from the client CIDR range. A portion of the addresses in the client CIDR range are also used to support the availability model of the Client VPN endpoint, and cannot be assigned to clients. You cannot change the client CIDR range after you create the Client VPN endpoint.

In general, we recommend that you specify a client CIDR range that contains twice the number of IP addresses (and therefore concurrent connections) that you plan to support on the Client VPN endpoint.

Number of associated subnets

When you associate a subnet (p. 59) with a Client VPN endpoint, you enable users to establish VPN sessions to the Client VPN endpoint. You can associate multiple subnets with a Client VPN endpoint for high availability, and to enable additional connection capacity.

The following are the number of supported concurrent VPN connections based on the number of subnet associations for the Client VPN endpoint.

Subnet associations	Supported number of connections
1	7,000
2	36,500
3	66,500
4	96,500
5	126,000

You cannot associate multiple subnets from the same Availability Zone with a Client VPN endpoint. Therefore, the number of subnet associations also depends on the number of Availability Zones that are available in an AWS Region.

For example, if you expect to support 8,000 VPN connections to your Client VPN endpoint, specify a minimum client CIDR range size of `/18` (16,384 IP addresses), and associate at least 2 subnets with the Client VPN endpoint.

If you're unsure what the number of expected VPN connections is for your Client VPN endpoint, we recommend that you specify a size `/16` CIDR block or larger.

For more information about the rules and limitations for working with client CIDR ranges and target networks, see Limitations and rules of Client VPN (p. 3).

For more information about quotas for your Client VPN endpoint, see AWS Client VPN quotas (p. 74).

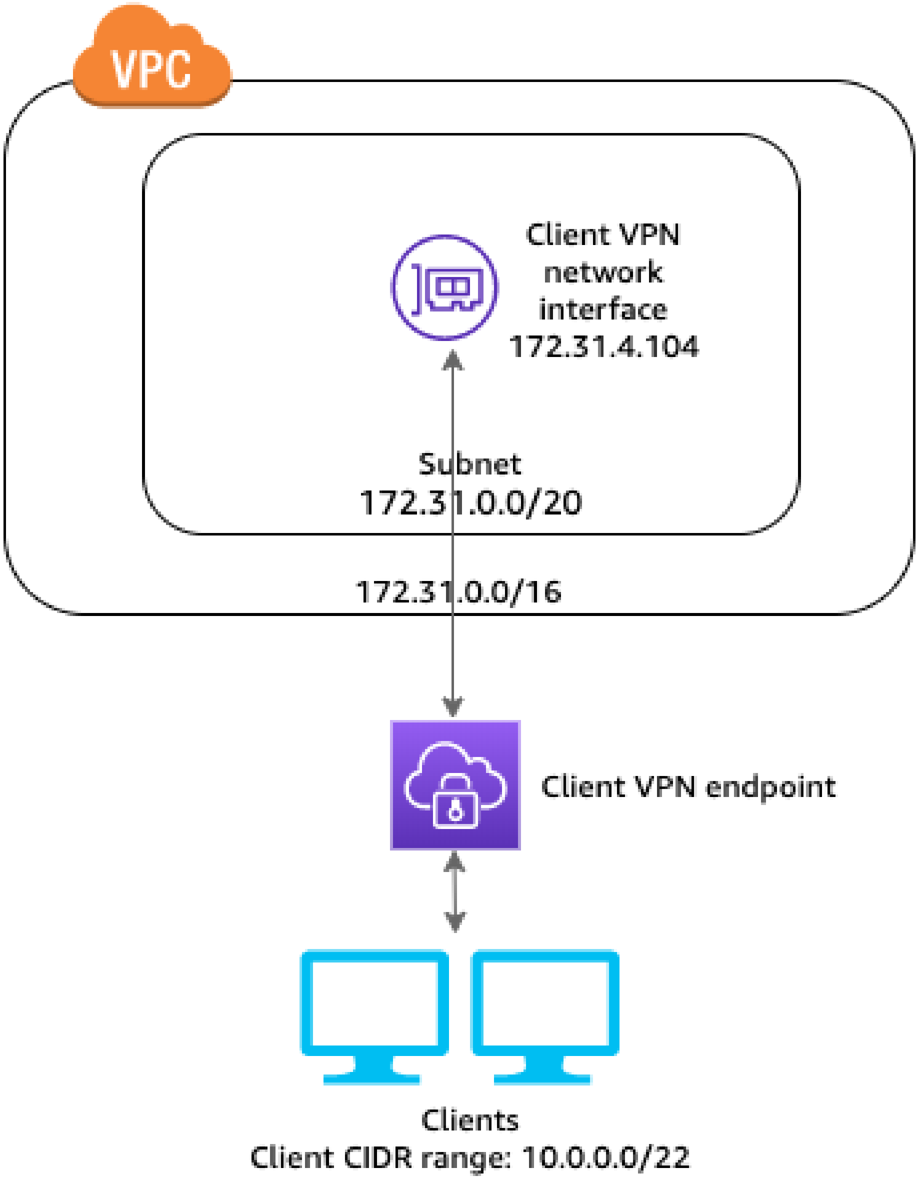
Scenarios and examples

This section provides examples for creating and configuring Client VPN access for your clients.

Contents

Access to a VPC

The configuration for this scenario includes a single target VPC. We recommend this configuration if you need to give clients access to the resources inside a single VPC only.


VPC
Client VPN network interface 172.31.4.104
Subnet 172.31.0.0/20
172.31.0.0/16
Client VPN endpoint
Clients
Client CIDR range: 10.0.0.0/22

Before you begin, do the following:

- Create or identify a VPC with at least one subnet. Identify the subnet in the VPC that you want to associate with the Client VPN endpoint and note its IPv4 CIDR ranges. For more information, see VPCs and Subnets in the *Amazon VPC User Guide*.
- Identify a suitable CIDR range for the client IP addresses that does not overlap with the VPC CIDR.
- Review the rules and limitations for Client VPN endpoints in Limitations and rules of Client VPN (p. 3).

To implement this configuration

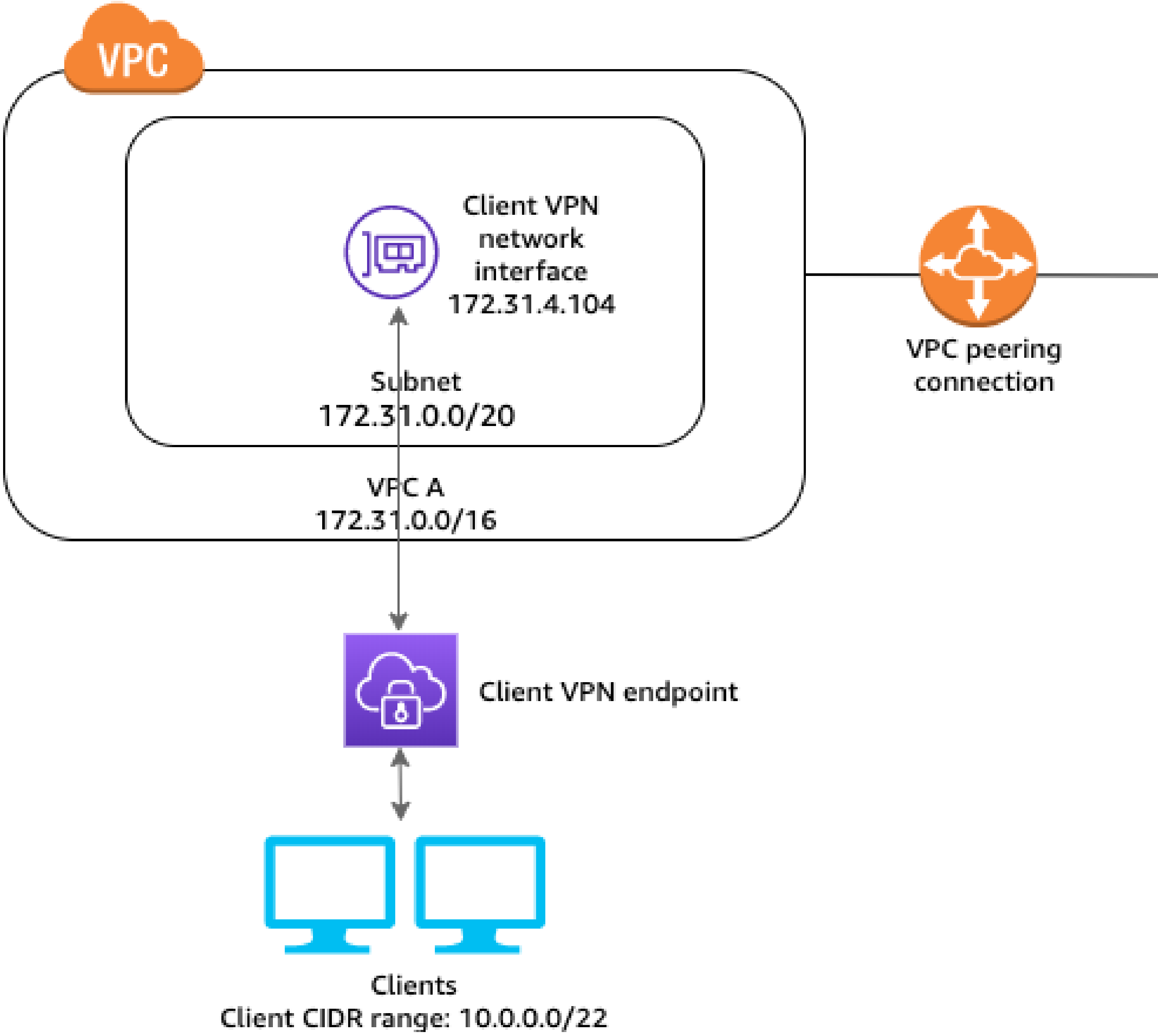
1. Create a Client VPN endpoint in the same Region as the VPC. To do this, perform the steps described in Create a Client VPN endpoint (p. 49).
2. Associate the subnet with the Client VPN endpoint. To do this, perform the steps described in Associate a target network with a Client VPN endpoint (p. 59) and select the subnet and the VPC you identified earlier.

3. Add an authorization rule to give clients access to the VPC. To do this, perform the steps described in Add an authorization rule to a Client VPN endpoint (p. 42), and for **Destination network**, enter the IPv4 CIDR range of the VPC.
4. Add a rule to your resources' security groups to allow traffic from the security group that was applied to the subnet association in step 2. For more information, see Security groups (p. 13).

Access to a peered VPC

The configuration for this scenario includes a target VPC (VPC A) that is peered with an additional VPC (VPC B). We recommend this configuration if you need to give clients access to the resources inside a target VPC and other VPCs that are peered with it (such as VPC B).

> **Note**
> The procedure for allowing access to a peered VPC outlined below, is only required if the Client VPN endpoint was configured for split-tunnel mode. In full-tunnel mode, access to the peered VPC would be allowed by default.



Before you begin, do the following:

- Create or identify a VPC with at least one subnet. Identify the subnet in the VPC that you want to associate with the Client VPN endpoint and note its IPv4 CIDR ranges. For more information, see VPCs and Subnets in the *Amazon VPC User Guide*.
- Identify a suitable CIDR range for the client IP addresses that does not overlap with the VPC CIDR.
- Review the rules and limitations for Client VPN endpoints in Limitations and rules of Client VPN (p. 3).

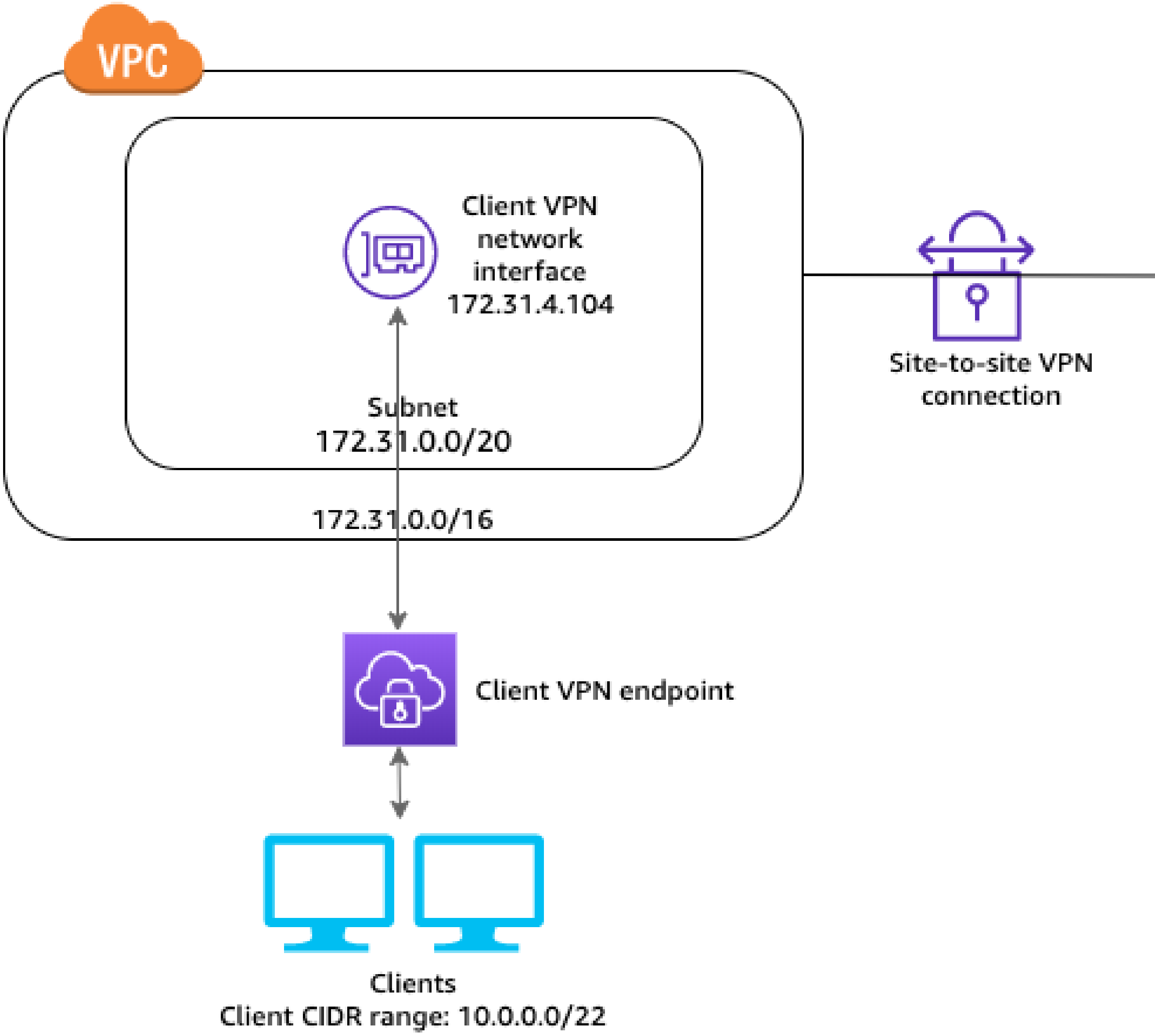
To implement this configuration

1. Establish the VPC peering connection between the VPCs. Follow the steps at Creating and accepting a VPC peering connection in the *Amazon VPC Peering Guide*.
2. Test the VPC peering connection. Confirm that instances in either VPC can communicate with each other as if they are within the same network. If the peering connection works as expected, continue to the next step.

3. Create a Client VPN endpoint in the same Region as the target VPC. In the preceding example, this is VPC A. Perform the steps described in Create a Client VPN endpoint (p. 49).
4. Associate the subnet you identified earlier with the Client VPN endpoint that you created. To do this, perform the steps described in Associate a target network with a Client VPN endpoint (p. 59) and select the subnet and the VPC.
5. Add an authorization rule to give clients access to the target VPC. To do this, perform the steps described in Add an authorization rule to a Client VPN endpoint (p. 42), and for **Destination network to enable** , enter the IPv4 CIDR range of the VPC.
6. Add a route to direct traffic to the peered VPC. In the preceding example, this is VPC B. To do this, perform the steps described in Create an endpoint route (p. 58); for **Route destination**, enter IPv4 CIDR range of the peered VPC, and for **Target VPC Subnet ID**, select the subnet you associated with the Client VPN endpoint.
7. Add an authorization rule to give clients access to peered VPC. To do this, perform the steps described in Add an authorization rule to a Client VPN endpoint (p. 42); for **Destination network**, enter IPv4 CIDR range of the peered VPC.
8. Add a rule to your resources' security groups in VPC A and VPC B to allow traffic from the security group that was applied to the subnet association in step 2. For more information, see Security groups (p. 13).

Access to an on-premises network

The configuration for this scenario includes access to an on-premises network only. We recommend this configuration if you need to give clients access to the resources inside an on-premises network only.


VPC
Client VPN network interface 172.31.4.104
Subnet 172.31.0.0/20
172.31.0.0/16
Site-to-site VPN connection
Client VPN endpoint
Clients
Client CIDR range: 10.0.0.0/22

Before you begin, do the following:

- Create or identify a VPC with at least one subnet. Identify the subnet in the VPC that you want to associate with the Client VPN endpoint and note its IPv4 CIDR ranges. For more information, see VPCs and Subnets in the *Amazon VPC User Guide*.
- Identify a suitable CIDR range for the client IP addresses that does not overlap with the VPC CIDR.
- Review the rules and limitations for Client VPN endpoints in Limitations and rules of Client VPN (p. 3).

To implement this configuration

1. Enable communication between the VPC and your own on-premises network over an AWS Site-to-Site VPN connection. To do this, perform the steps described in Getting started in the *AWS Site-to-Site VPN User Guide*.

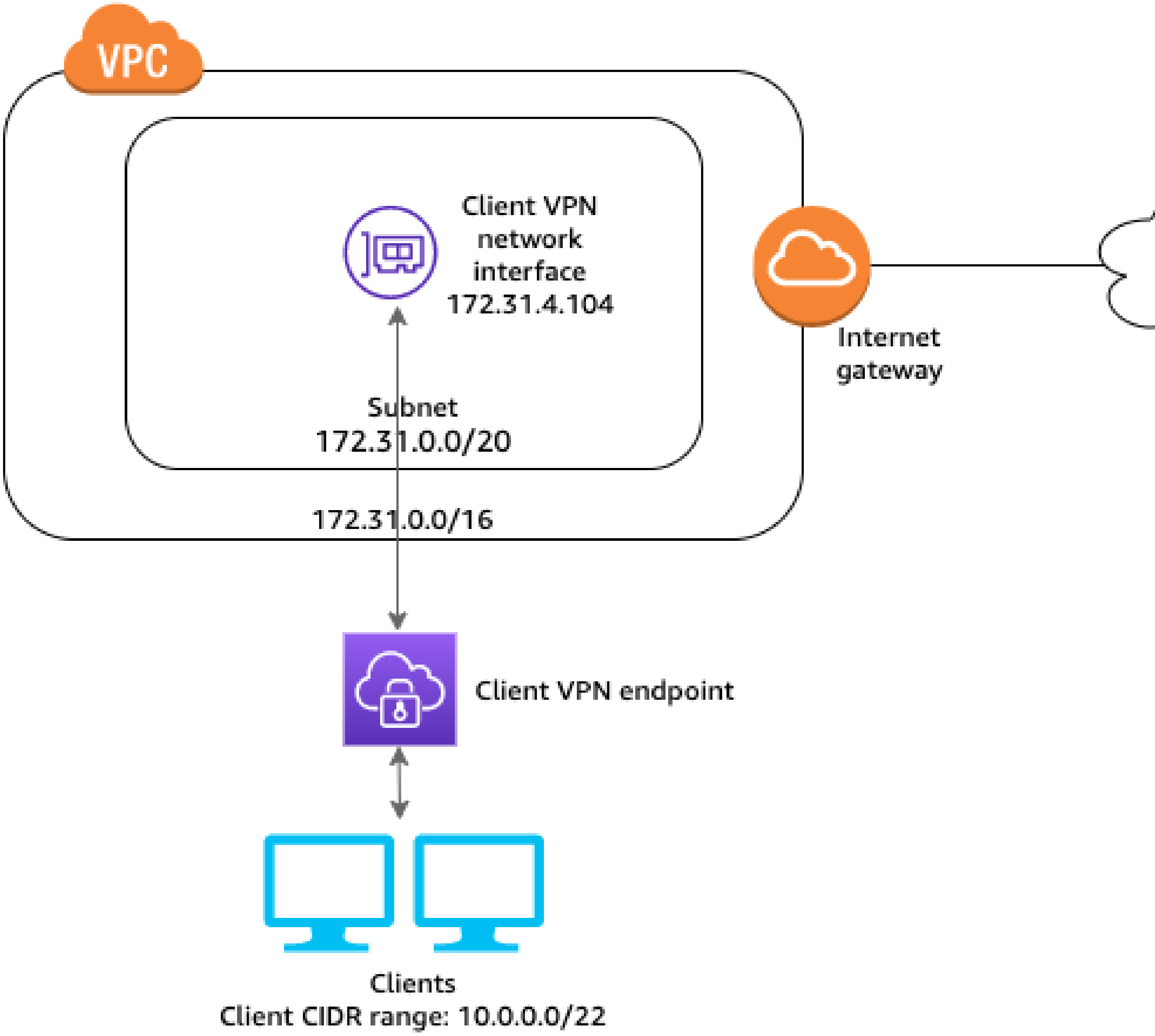
> **Note**
> Alternatively, you can implement this scenario by using an AWS Direct Connect connection between your VPC and your on-premises network. For more information, see the AWS Direct Connect User Guide.

2. Test the AWS Site-to-Site VPN connection you created in the previous step. To do this, perform the steps described in Testing the Site-to-Site VPN connection in the *AWS Site-to-Site VPN User Guide*. If the VPN connection is functioning as expected, continue to the next step.
3. Create a Client VPN endpoint in the same Region as the VPC. To do this, perform the steps described in Create a Client VPN endpoint (p. 49).
4. Associate the subnet that you identified earlier with the Client VPN endpoint. To do this, perform the steps described in Associate a target network with a Client VPN endpoint (p. 59) and select the VPC and the subnet.
5. Add a route that allows access to the AWS Site-to-Site VPN connection. To do this, perform the steps described in Create an endpoint route (p. 58); for **Route destination**, enter the IPv4 CIDR range of the AWS Site-to-Site VPN connection, and for **Target VPC Subnet ID**, select the subnet you associated with the Client VPN endpoint.
6. Add an authorization rule to give clients access to the AWS Site-to-Site VPN connection. To do this, perform the steps described in Add an authorization rule to a Client VPN endpoint (p. 42); for **Destination network**, enter the AWS Site-to-Site VPN connection IPv4 CIDR range.

Access to the internet

The configuration for this scenario includes a single target VPC and access to the internet. We recommend this configuration if you need to give clients access to the resources inside a single target VPC and allow access to the internet.

If you completed the Getting started with Client VPN (p. 35) tutorial, then you've already implemented this scenario.



Before you begin, do the following:

- Create or identify a VPC with at least one subnet. Identify the subnet in the VPC that you want to associate with the Client VPN endpoint and note its IPv4 CIDR ranges. For more information, see VPCs and Subnets in the *Amazon VPC User Guide*.
- Identify a suitable CIDR range for the client IP addresses that does not overlap with the VPC CIDR.
- Review the rules and limitations for Client VPN endpoints in Limitations and rules of Client VPN (p. 3).

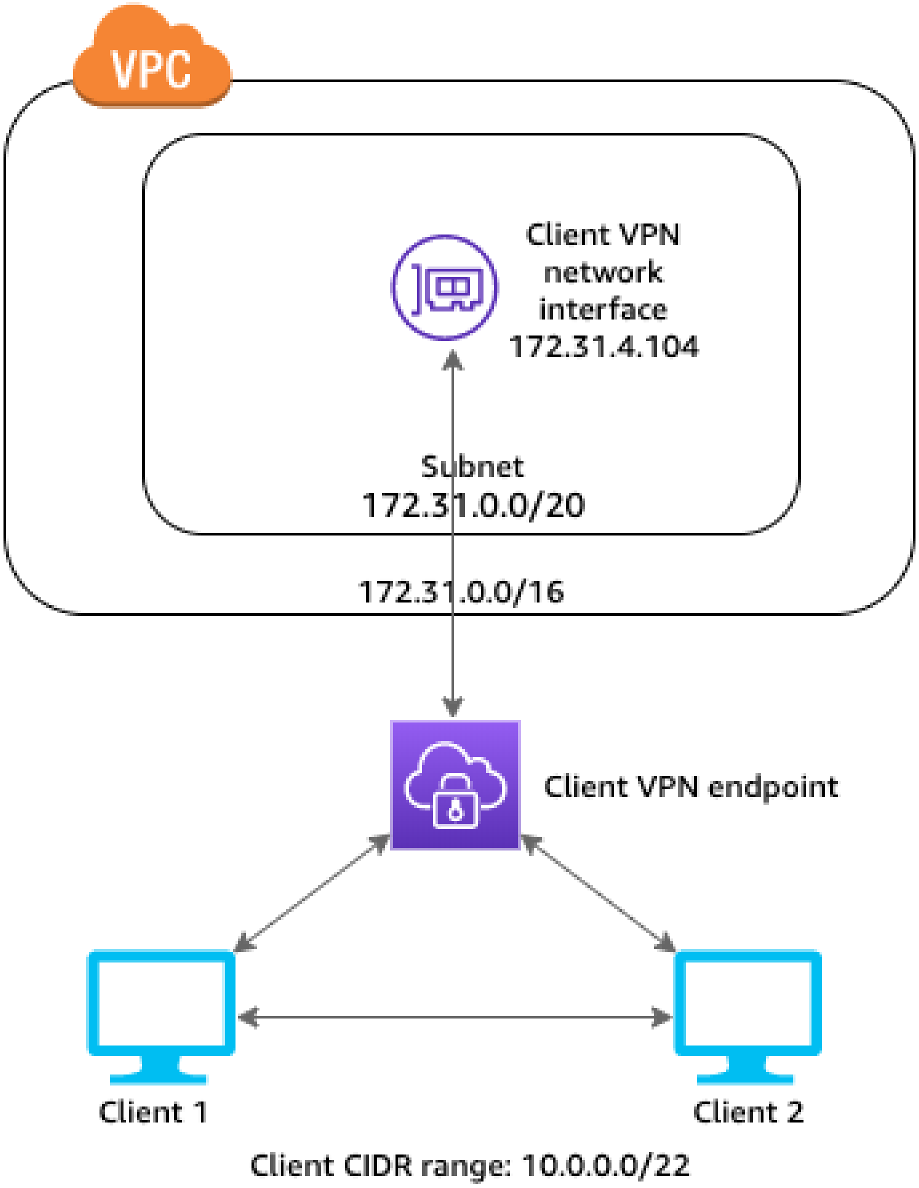
To implement this configuration

1. Ensure that the security group that you'll use for the Client VPN endpoint allows outbound traffic to the internet. To do this, add outbound rules that allow traffic to 0.0.0.0/0 for HTTP and HTTPS traffic.
2. Create an internet gateway and attach it to your VPC. For more information, see Creating and Attaching an Internet Gateway in the *Amazon VPC User Guide*.

3. Make your subnet public by adding a route to the internet gateway to its route table. In the VPC console, choose **Subnets**, select the subnet you intend to associate with the Client VPN endpoint, choose **Route Table**, and then choose the route table ID. Choose **Actions**, choose **Edit routes**, and choose **Add route**. For **Destination**, enter `0.0.0.0/0`, and for **Target**, choose the internet gateway from the previous step.
4. Create a Client VPN endpoint in the same Region as the VPC. To do this, perform the steps described in Create a Client VPN endpoint (p. 49).
5. Associate the subnet that you identified earlier with the Client VPN endpoint. To do this, perform the steps described in Associate a target network with a Client VPN endpoint (p. 59) and select the VPC and the subnet.
6. Add an authorization rule to give clients access to the VPC. To do this, perform the steps described in Add an authorization rule to a Client VPN endpoint (p. 42); and for **Destination network to enable** , enter the IPv4 CIDR range of the VPC.
7. Add a route that enables traffic to the internet. To do this, perform the steps described in Create an endpoint route (p. 58); for **Route destination**, enter `0.0.0.0/0`, and for **Target VPC Subnet ID**, select the subnet you associated with the Client VPN endpoint.
8. Add an authorization rule to give clients access to the internet. To do this, perform the steps described in Add an authorization rule to a Client VPN endpoint (p. 42); for **Destination network**, enter `0.0.0.0/0`.
9. Ensure that the security groups for the resources in your VPC have a rule that allows access from the security group associated with the Client VPN endpoint. This enables your clients to access the resources in your VPC.

Client-to-client access

The configuration for this scenario enables clients to access a single VPC, and enables clients to route traffic to each other. We recommend this configuration if the clients that connect to the same Client VPN endpoint also need to communicate with each other. Clients can communicate with each other using the unique IP address that's assigned to them from the client CIDR range when they connect to the Client VPN endpoint.


VPC
Client VPN network interface 172.31.4.104
Subnet 172.31.0.0/20
172.31.0.0/16
Client VPN endpoint
Client 1
Client 2
Client CIDR range: 10.0.0.0/22

Before you begin, do the following:

- Create or identify a VPC with at least one subnet. Identify the subnet in the VPC that you want to associate with the Client VPN endpoint and note its IPv4 CIDR ranges. For more information, see VPCs and Subnets in the *Amazon VPC User Guide*.
- Identify a suitable CIDR range for the client IP addresses that does not overlap with the VPC CIDR.
- Review the rules and limitations for Client VPN endpoints in Limitations and rules of Client VPN (p. 3).

Note
Network-based authorization rules using Active Directory groups or SAML-based IdP groups are not supported in this scenario.

To implement this configuration

1. Create a Client VPN endpoint in the same Region as the VPC. To do this, perform the steps described in Create a Client VPN endpoint (p. 49).
2. Associate the subnet that you identified earlier with the Client VPN endpoint. To do this, perform the steps described in Associate a target network with a Client VPN endpoint (p. 59) and select the VPC and the subnet.
3. Add a route to the local network in the route table. To do this, perform the steps described in Create an endpoint route (p. 58). For **Route destination**, enter the client CIDR range, and for **Target VPC Subnet ID**, specify `local`.
4. Add an authorization rule to give clients access to the VPC. To do this, perform the steps described in Add an authorization rule to a Client VPN endpoint (p. 42). For **Destination network to enable** , enter the IPv4 CIDR range of the VPC.
5. Add an authorization rule to give clients access to the client CIDR range. To do this, perform the steps described in Add an authorization rule to a Client VPN endpoint (p. 42). For **Destination network to enable**, enter the client CIDR range.

Restrict access to your network

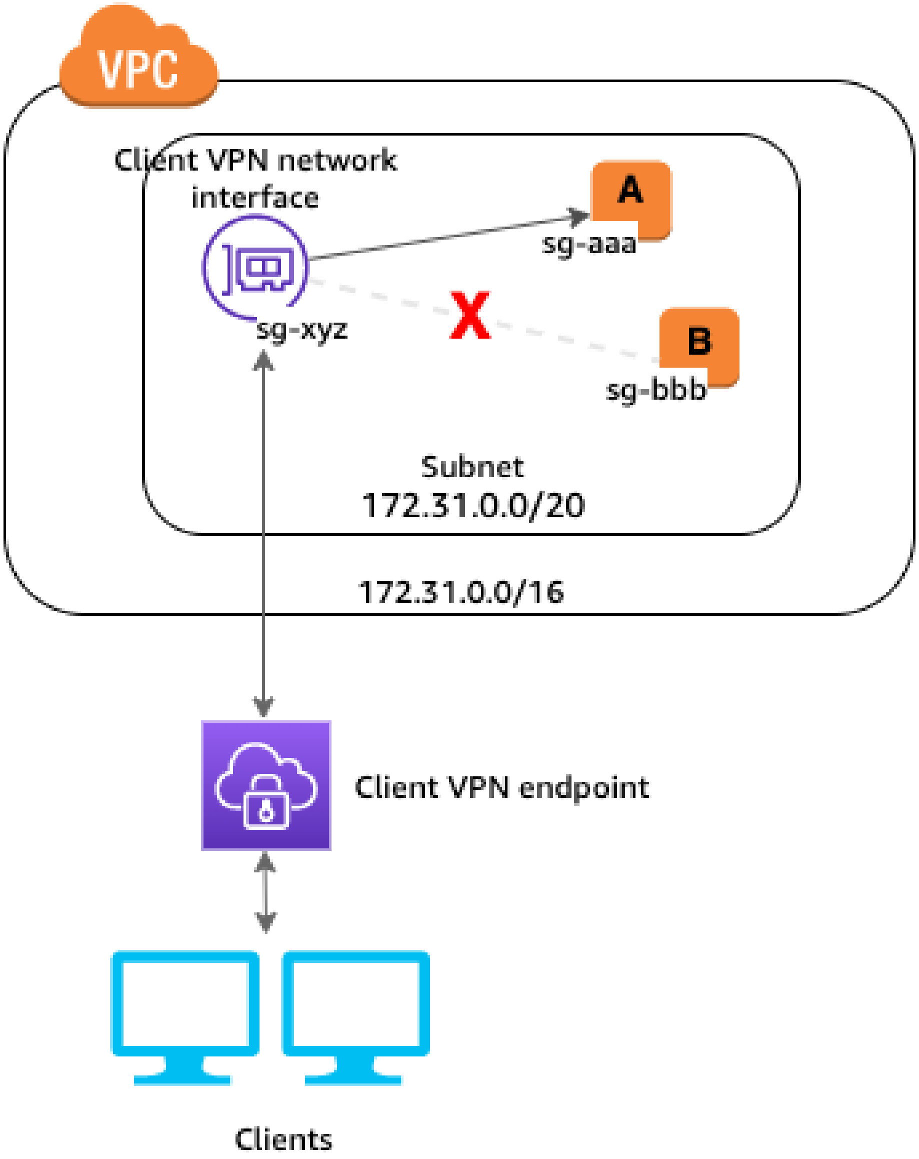
You can configure your Client VPN endpoint to restrict access to specific resources in your VPC. For user-based authentication, you can also restrict access to parts of your network, based on the user group that accesses the Client VPN endpoint.

Restrict access using security groups

You can grant or deny access to specific resources in your VPC by adding or removing security group rules that reference the security group that was applied to the target network association (the Client VPN security group). This configuration expands on the scenario described in Access to a VPC (p. 22). This configuration is applied in addition to the authorization rule configured in that scenario.

To grant access to a specific resource, identify the security group that's associated with the instance on which your resource is running. Then, create a rule that allows traffic from the Client VPN security group.

In the following example, `sg-xyz` is the Client VPN security group, security group `sg-aaa` is associated with instance A, and security group `sg-bbb` is associated with instance B. You add a rule to `sg-aaa` that allows access from `sg-xyz`, therefore, clients can access your resources in instance A. Security group `sg-bbb` does not have a rule that allows access from `sg-xyz` or the Client VPN network interface. Clients cannot access the resources in instance B.



Before you begin, check if the Client VPN security group is associated with other resources in your VPC. If you add or remove rules that reference the Client VPN security group, you might grant or deny access for the other associated resources too. To prevent this, use a security group that is specifically created for use with your Client VPN endpoint.

To create a security group rule

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Security Groups**.
3. Choose the security group that's associated with the instance on which your resource is running.
4. Choose **Actions**, **Edit inbound rules**.
5. Choose **Add rule**, and then do the following:

   - For **Type**, choose **All traffic**, or a specific type of traffic that you want to allow.
   - For **Source**, choose **Custom**, and then enter or choose the ID of the Client VPN security group.

6. Choose **Save rules**

To remove access to a specific resource, check the security group that's associated with the instance on which your resource is running. If there is a rule that allows traffic from the Client VPN security group, delete it.

To check your security group rules

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Inbound Rules**.
4. Review the list of rules. If there is a rule where **Source** is the Client VPN security group, choose **Edit Rules**, and choose **Delete** (the x icon) for the rule. Choose **Save rules**.

Restrict access based on user groups

If your Client VPN endpoint is configured for user-based authentication, you can grant specific groups of users access to specific parts of your network. To do this, complete the following steps:

1. Configure users and groups in AWS Directory Service or your IdP. For more information, see the following topics:
   - Active Directory authentication (p. 6)
   - Requirements and considerations for SAML-based federated authentication (p. 12)
2. Create an authorization rule for your Client VPN endpoint that allows a specified group access to all or part of your network. For more information, see Authorization rules (p. 42).

If your Client VPN endpoint is configured for mutual authentication, you cannot configure user groups. When you create an authorization rule, you must grant access to all users. To enable specific groups of users access to specific parts of your network, you can create multiple Client VPN endpoints. For example, for each group of users that accesses your network, do the following:

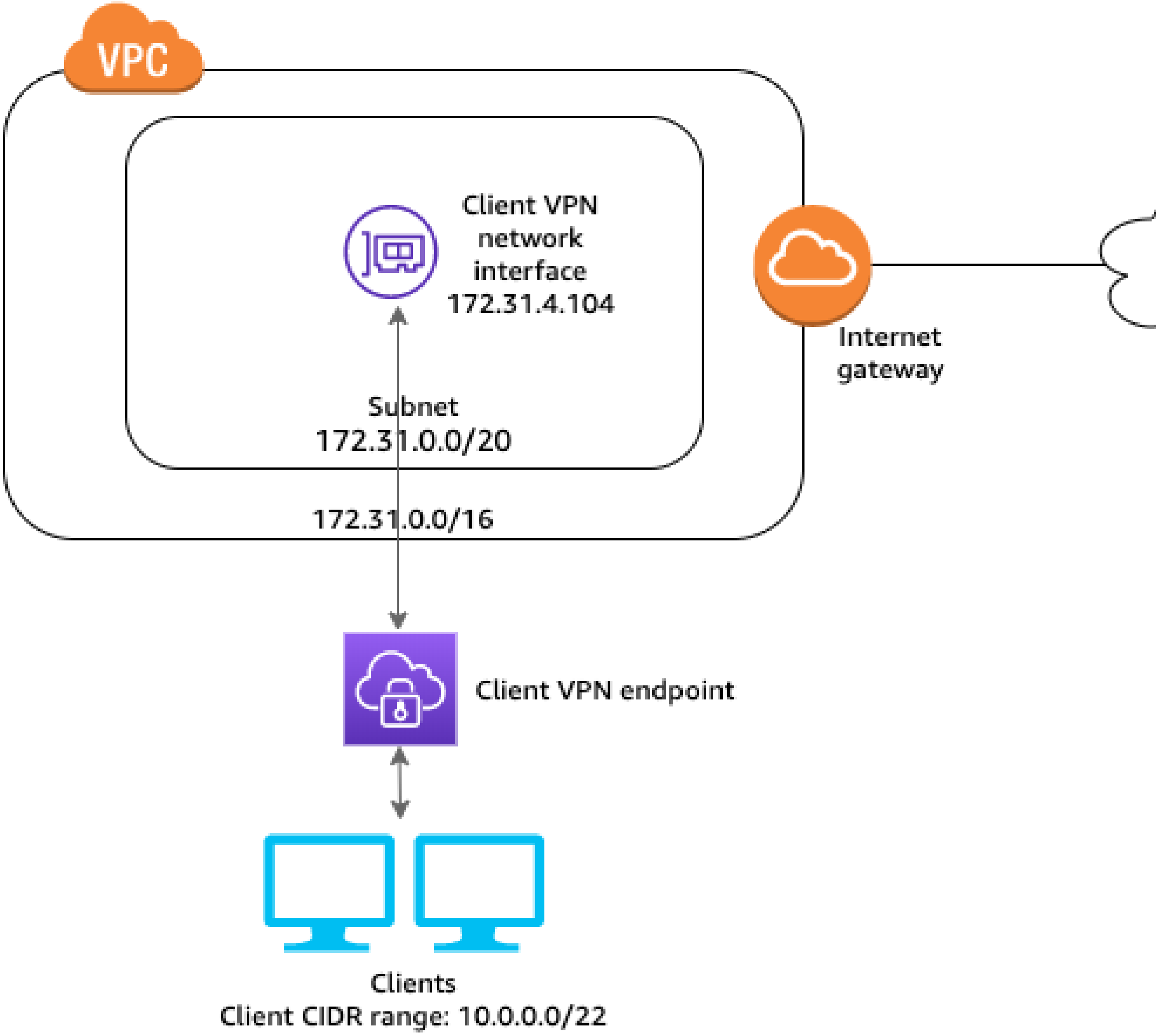
1. Create a set of server and client certificates and keys for that group of users. For more information, see Mutual authentication (p. 6).
2. Create a Client VPN endpoint. For more information, see Create a Client VPN endpoint (p. 49).
3. Create an authorization rule that grants access to all or part of your network. For example, for a Client VPN endpoint that is used by administrators, you might create an authorization rule that grants access to the entire network. For more information, see Add an authorization rule to a Client VPN endpoint (p. 42).

Getting started with Client VPN

In this tutorial you will create a Client VPN endpoint that does the following:

- Provides all clients with access to a single VPC.
- Provides all clients with access to the internet.
- Uses mutual authentication (p. 6).

The following diagram represents the configuration of your VPC and Client VPN endpoint after you've completed this tutorial.



Steps

Prerequisites

Before you begin this getting started tutorial, make sure that you have the following:

- The permissions required to work with Client VPN endpoints.
- The permissions required to import certificates into AWS Certificate Manager.
- A VPC with at least one subnet and an internet gateway. The route table that's associated with your subnet must have a route to the internet gateway.

Step 1: Generate server and client certificates and keys

This tutorial uses mutual authentication. With mutual authentication, Client VPN uses certificates to perform authentication between clients and the Client VPN endpoint. You will need to have a server certificate and key, and at least one client certificate and key. At minimum, the server certificate will need to be imported into AWS Certificate Manager (ACM) and specified when you create the Client VPN endpoint. Importing the client certificate into ACM is optional.

If you don't already have certificates to use for this purpose, they can be created using the OpenVPN easy-rsa utility. For detailed steps to generate the server and client certificates and keys using the OpenVPN easy-rsa utility, and import them into ACM see Mutual authentication (p. 6).

Step 2: Create a Client VPN endpoint

The Client VPN endpoint is the resource that you create and configure to enable and manage client VPN sessions. It's the termination point for all client VPN sessions.

To create a Client VPN endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints** and then choose **Create Client VPN endpoint**.
3. (Optional) Provide a name tag and description for the Client VPN endpoint.
4. For **Client IPv4 CIDR**, specify an IP address range, in CIDR notation, from which to assign client IP addresses. For example, `10.0.0.0/22`.

> **Note**
> The address range cannot overlap with the target network address range, the VPC address range, or any of the routes that will be associated with the Client VPN endpoint. The client address range must be at minimum /22 and not greater than /12 CIDR block size. You cannot change the client address range after you create the Client VPN endpoint.

5. For **Server certificate ARN**, select the ARN of the server certificate that you generated in Step 1 (p. 36).

   > **Note**
   > The server certificate must be provisioned with or imported into AWS Certificate Manager (ACM) in the same AWS Region.

6. Under **Authentication options**, choose **Use mutual authentication**, and then for **Client certificate ARN**, select the ARN of the certificate you want to use as the client certificate.

   > **Note**
   > If the server and client certificates are signed by the same certificate authority (CA), you have the option of specifying the server certificate ARN for *both* the client and server certificates. In this scenario, any client certificate that corresponds with the server certificate can be used to authenticate.

7. Keep the rest of the default settings, and choose **Create Client VPN endpoint**.

After you create the Client VPN endpoint, its state is `pending-associate`. Clients can only establish a VPN connection after you associate at least one target network.

For more information about the other options that you can specify when creating a Client VPN endpoint, see Create a Client VPN endpoint (p. 49).

Step 3: Associate a target network

To allow clients to establish a VPN session, you associate a target network with the Client VPN endpoint. A target network is a subnet in a VPC.

To associate a target network with the Client VPN endpoint

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint that you created in the preceding procedure, and then choose **Target network associations**, **Associate target network**.
4. For **VPC**, choose the VPC in which the subnet is located.
5. For **Choose a subnet to associate**, choose the subnet to associate with the Client VPN endpoint.
6. Choose **Associate target network**.

   > **Note**
   > If authorization rules allow it, one subnet association is enough for clients to access a VPC's entire network. You can associate additional subnets to provide high availability in case one of the Availability Zones goes down.

When you associate the first subnet with the Client VPN endpoint, the following happens:

- The state of the Client VPN endpoint changes to `available`. Clients can now establish a VPN connection, but they cannot access any resources in the VPC until you add the authorization rules.
- The local route of the VPC is automatically added to the Client VPN endpoint route table.

- The VPC's default security group is automatically applied for the Client VPN endpoint.

Step 4: Add an authorization rule for the VPC

For clients to access the VPC, there needs to be a route to the VPC in the Client VPN endpoint's route table and an authorization rule. The route was already added automatically in the previous step. For this tutorial, we want to grant all users access to the VPC.

To add an authorization rule for the VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint to which to add the authorization rule. Choose **Authorization rules**, and then choose **Add authorization rule**.
4. For **Destination network to enable access**, enter the CIDR of the network for which you want to allow access. For example, to allow access to the entire VPC, specify the IPv4 CIDR block of the VPC.
5. For **Grant access to**, choose **Allow access to all users**.
6. (Optional) For **Description**, enter a brief description of the authorization rule.
7. Choose **Add authorization rule**.

Step 5: Provide access to the internet

You can provide access to additional networks connected to the VPC, such as AWS services, peered VPCs, on-premises networks, and the internet. For each additional network, you add a route to the network in the Client VPN endpoint's route table and configure an authorization rule to give clients access.

For this tutorial, we want to grant all users access to the internet and also to the VPC. You've already configured access to the VPC, so this step is for access to the internet.

To provide access to the internet

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint that you created for this tutorial. Choose **Route Table**, and then choose **Create Route**.
4. For **Route destination**, enter `0.0.0.0/0`. For **Subnet ID for target network association**, specify the ID of the subnet through which to route traffic.
5. Choose **Create Route**.
6. Choose **Authorization rules**, and then choose **Add authorization rule**.
7. For **Destination network to enable access**, enter `0.0.0.0/0`, and choose **Allow access to all users**.
8. Choose **Add authorization rule**.

Step 6: Verify security group requirements

In this tutorial, no security groups were specified during the creation of the Client VPN endpoint in Step 2. That means that the default security group for the VPC is automatically applied to the Client VPN endpoint when a target network is associated. As a result, the default security group for the VPC should now be associated with the Client VPN endpoint.

Verify the following security group requirements

- That the security group associated with subnet you are routing traffic through (in this case the default VPC security group) allows outbound traffic to the internet. To do this, add an outbound rule that allows all traffic to destination `0.0.0.0/0`.
- That the security groups for the resources in your VPC have a rule that allows access from the security group that's applied to the Client VPN endpoint (in this case the default VPC security group). This enables your clients to access the resources in your VPC.

For more information, see Security groups (p. 13).

Step 7: Download the Client VPN endpoint configuration file

The next step is to download and prepare the Client VPN endpoint configuration file. The configuration file includes the Client VPN endpoint details and certificate information required to establish a VPN connection. You provide this file to the end users who need to connect to the Client VPN endpoint. The end user uses the file to configure their VPN client application.

To download and prepare the Client VPN endpoint configuration file

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint that you created for this tutorial, and choose **Download client configuration**.
4. Locate the client certificate and key that were generated in Step 1 (p. 36). The client certificate and key can be found in the following locations in the cloned OpenVPN easy-rsa repo:

   - Client certificate — `easy-rsa/easyrsa3/pki/issued/client1.domain.tld.crt`
   - Client key — `easy-rsa/easyrsa3/pki/private/client1.domain.tld.key`
5. Open the Client VPN endpoint configuration file using your preferred text editor. Add `<cert></cert>` and `<key></key>` tags to the file. Place the contents of the client certificate and the contents of the private key between the corresponding tags, as such:

```
<cert>
Contents of client certificate (.crt) file
</cert>

<key>
Contents of private key (.key) file
</key>
```

6. Locate the line that specifies the Client VPN endpoint DNS name, and prepend a random string to it so that the format is *`random_string.displayed_DNS_name`*. For example:

   - Original DNS name: `cvpn-endpoint-0102bc4c2eEXAMPLE.prod.clientvpn.us-west-2.amazonaws.com`
   - Modified DNS name: `asdfa.cvpn-endpoint-0102bc4c2eEXAMPLE.prod.clientvpn.us-west-2.amazonaws.com`

> **Note**
>
> We recommend that you always use the DNS name provided for the Client VPN endpoint in your configuration file, as described. The IP addresses that the DNS name will resolve to are subject to change.

7. Save and close the Client VPN endpoint configuration file.
8. Distribute the Client VPN endpoint configuration file to your end users.

For more information about the Client VPN endpoint configuration file, see Export and configure the client configuration file (p. 55).

Step 8: Connect to the Client VPN endpoint

You can connect to the Client VPN endpoint using the AWS provided client or another OpenVPN-based client application and the configuration file that you just created. For more information, see the AWS Client VPN User Guide.

Working with Client VPN

You can work with Client VPN using the Amazon VPC console or the AWS CLI.

Contents

Access the self-service portal

If you enabled the self-service portal for your Client VPN endpoint, you can provide your clients with a self-service portal URL. Clients can access the portal in a web browser, and use their user-based credentials to log in. In the portal, clients can download the Client VPN endpoint configuration file and they can download the latest version of the AWS provided client.

The following rules apply:

- The self-service portal is not available for clients that authenticate using mutual authentication.
- The configuration file that's available in the self-service portal is the same configuration file that you export using the Amazon VPC console or AWS CLI. If you need to customize the configuration file before distributing it to clients, you must distribute the customized file to clients yourself.
- You must enable the self-service portal option for your Client VPN endpoint, or clients cannot access the portal. If this option is not enabled, you can modify your Client VPN endpoint to enable it.

After you have enabled the self-service portal option, provide your clients with one of the following URLs:

- `https://self-service.clientvpn.amazonaws.com/`

  If clients access the portal using this URL, they must enter the ID of the Client VPN endpoint before they can log in.
- `https://self-service.clientvpn.amazonaws.com/endpoints/<endpoint-id>`

  Replace *`<endpoint-id>`* in the preceding URL with the ID of your Client VPN endpoint, for example, `cvpn-endpoint-0123456abcd123456`.

You can also view the URL for the self-service portal in the output of the describe-client-vpn-endpoints AWS CLI command. Alternatively, the URL is available in the **Details** tab on the **Client VPN Endpoints** page in the Amazon VPC console.

For more information about configuring the self-service portal for use with federated authentication, see Support for the self-service portal (p. 13).

Authorization rules

Authorization rules act as firewall rules that grant access to networks. By adding authorization rules, you grant specific clients access to the specified network. You should have an authorization rule for each network you want to grant access to. You can add authorization rules to a Client VPN endpoint using the console and the AWS CLI.

> **Note**
> Client VPN uses longest prefix matching when evaluating authorization rules. See the troubleshooting topic Authorization rules for Active Directory groups not working as expected (p. 77) and Route priority in the *Amazon VPC User Guide* for more details.

Contents

Add an authorization rule to a Client VPN endpoint

To add an authorization rule to a Client VPN endpoint using AWS Management Console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint to which to add the authorization rule, choose **Authorization rules**, and choose **Add authorization rule**.
4. For **Destination network to enable access**, enter the IP address, in CIDR notation, of the network that you want users to access (for example, the CIDR block of your VPC).
5. Specify which clients are allowed to access the specified network. For **For grant access to**, do one of the following:

   - To grant access to all clients, choose **Allow access to all users**.
   - To restrict access to specific clients, choose **Allow access to users in a specific access group**, and then for **Access group ID**, enter the ID for the group to grant access to. For example, the security identifier (SID) of an Active Directory group, or the ID/name of a group defined in a SAML-based identity provider (IdP).
     - (Active Directory) To get the SID, you can use the Microsoft Powershell Get-ADGroup cmdlet, for example:

       ```
       Get-ADGroup -Filter 'Name -eq "<Name of the AD Group>"'
       ```

       Alternatively, open the Active Directory Users and Computers tool, view the properties for the group, go to the Attribute Editor tab, and get the value for `objectSID`. If necessary, first choose **View**, **Advanced Features** to enable the Attribute Editor tab.
     - (SAML-based federated authentication) The group ID/name should match the group attribute information that is returned in the SAML assertion.
6. For **Description**, enter a brief description of the authorization rule.
7. Choose **Add authorization rule**.

To add an authorization rule to a Client VPN endpoint (AWS CLI)

Use the authorize-client-vpn-ingress command.

Remove an authorization rule from a Client VPN endpoint

By deleting an authorization rule, you remove access to the specified network.

You can remove authorization rules from a Client VPN endpoint using the console and the AWS CLI.

To remove an authorization rule from a Client VPN endpoint (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint to which the authorization rule is added and choose **Authorization rules**.
4. Select the authorization rule to delete, choose **Remove authorization rule**, and choose **Remove authorization rule**.

To remove an authorization rule from a Client VPN endpoint (AWS CLI)

Use the revoke-client-vpn-ingress command.

View authorization rules

You can view authorization rules for a specific Client VPN endpoint using the console and the AWS CLI.

To view authorization rules (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint for which to view authorization rules and choose **Authorization rules**.

To view authorization rules (AWS CLI)

Use the describe-client-vpn-authorization-rules command.

Client certificate revocation lists

You can use client certificate revocation lists to revoke access to a Client VPN endpoint for specific client certificates.

> **Note**
> For more information about generating the server and client certificates and keys, see Mutual authentication (p. 6)

For more information about the number of entries you can add to a client certificate revocation list, see Client VPN quotas (p. 74).

Contents

Generate a client certificate revocation list

Linux/macOS

In the following procedure, you generate a client certificate revocation list using the OpenVPN easy-rsa command line utility.

To generate a client certificate revocation list using OpenVPN easy-rsa

1. Clone the OpenVPN easy-rsa repo to your local computer.

```
$ git clone https://github.com/OpenVPN/easy-rsa.git
```

2. Navigate into the `easy-rsa/easyrsa3` folder in your local repo.

```
$ cd easy-rsa/easyrsa3
```

3. Revoke the client certificate and generate the client revocation list.

```
$ ./easyrsa revoke client_certificate_name
$ ./easyrsa gen-crl
```

   Type `yes` when prompted.

Windows

The following procedure uses the OpenVPN software to generate a client revocation list. It assumes that you followed the steps for using the OpenVPN software (p. 6) to generate the client and server certificates and keys.

To generate a client certificate revocation list using EasyRSA version 3.x.x

1. Open a command prompt and navigate to the EasyRSA-3.x.x directory, which will depend on where it is installed on your system.

```
C:\> cd c:\Users\windows\EasyRSA-3.x.x
```

2. Run the "EasyRSA-Start.bat" file to start the EasyRSA shell.

```
C:\> .\EasyRSA-Start.bat
```

3. In the EasyRSA shell, revoke the client certificate.

```
# ./easyrsa revoke client_certificate_name
```

4. Type "yes" when prompted.
5. Generate the client revocation list.

```
# ./easyrsa gen-crl
```

6. The client revocation list will be created in the following location:

```
c:\Users\windows\EasyRSA-3.x.x\pki\crl.pem
```

To generate a client certificate revocation list using previous EasyRSA versions

1. Open a command prompt and navigate to the OpenVPN directory.

```
C:\> cd \Program Files\OpenVPN\easy-rsa
```

2. Run the `vars.bat` file.

```
C:\> vars
```

3. Revoke the client certificate and generate the client revocation list.

```
C:\> revoke-full client_certificate_name
C:\> more crl.pem
```

Import a client certificate revocation list

You must have a client certificate revocation list file to import. For more information about generating a client certificate revocation list, see Generate a client certificate revocation list (p. 44).

You can import a client certificate revocation list using the console and the AWS CLI.

To import a client certificate revocation list (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint for which to import the client certificate revocation list.
4. Choose **Actions**, and choose **Import Client Certificate CRL**.
5. For **Certificate Revocation List**, enter the contents of the client certificate revocation list file, and choose **Import client certificate CRL**.

To import a client certificate revocation list (AWS CLI)

Use the import-client-vpn-client-certificate-revocation-list command.

```
$ aws ec2 import-client-vpn-client-certificate-revocation-list --certificate-revocation-list file:path_to_CRL_file --client-vpn-endpoint-id endpoint_id --region region
```

Export a client certificate revocation list

You can export client certificate revocation lists using the console and the AWS CLI.

To export a client certificate revocation list (console)

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

3. Select the Client VPN endpoint for which to export the client certificate revocation list.
4. Choose **Actions**, choose **Export Client Certificate CRL**, and choose **Export Client Certificate CRL**.

To export a client certificate revocation (AWS CLI)

Use the export-client-vpn-client-certificate-revocation-list command.

Client connections

Connections are VPN sessions that have been established by clients. A connection is established when a client successfully connects to a Client VPN endpoint.

Contents

View client connections

You can view client connections using the console and the AWS CLI. The connection information includes the IP address that's assigned from the client CIDR range.

To view client connections (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint for which to view client connections.
4. Choose the **Connections** tab. The **Connections** tab lists all active and terminated client connections.

To view client connections (AWS CLI)

Use the describe-client-vpn-connections command.

Terminate a client connection

When you terminate a client connection, the VPN session ends.

You can terminate client connections using the console and the AWS CLI.

To terminate a client connection (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint to which the client is connected and choose **Connections**.
4. Select the connection to terminate, choose **Terminate Connection**, and choose **Terminate Connection**.

To terminate a client connection (AWS CLI)

Use the terminate-client-vpn-connections command.

Client login banner

AWS Client VPN provides the option to display a text banner on AWS provided Client VPN desktop applications when a VPN session is established. You can define the contents of the text banner to meet your regulatory and compliance needs. A maximum of 1400, UTF-8 encoded characters can be used.

> **Note**
> When a client login banner has been enabled, it will be displayed on newly created VPN sessions only. Existing VPN sessions are not interrupted, though the banner will be displayed when an existing session is re-established.

See Release notes for the AWS provided client in the *AWS Client VPN User Guide* for details on client desktop applications.

Contents

Configure a client login banner during creation of a Client VPN endpoint

For detailed steps to enable a client login banner during creation of a Client VPN endpoint, see Create a Client VPN endpoint (p. 49).

Configure a client login banner for an existing Client VPN endpoint

Use the following steps to configure a client login banner for an existing Client VPN endpoint.

Enable client login banner on a Client VPN endpoint (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint that you want to modify, choose **Actions**, and then choose **Modify Client VPN Endpoint**.
4. Scroll down the page to the **Other parameters** section.
5. Turn on **Enable client login banner**.
6. For **Client login banner text**, enter the text that will be displayed in a banner on AWS provided clients when a VPN session is established. Use UTF-8 encoded characters only, with a maximum of 1400 characters allowed.
7. Choose **Modify Client VPN endpoint**.

Enable client login banner on a Client VPN endpoint (AWS CLI)

Use the modify-client-vpn-endpoint command.

Deactivate a client login banner for an existing Client VPN endpoint

Use the following steps to deactivate a client login banner for an existing Client VPN endpoint.

Deactivate client login banner on a Client VPN endpoint (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint that you want to modify, choose **Actions**, and then choose **Modify Client VPN endpoint**.
4. Scroll down the page to the **Other parameters** section.
5. Turn off **Enable client login banner?**.
6. Choose **Modify Client VPN endpoint**.

Deactivate client login banner on a Client VPN endpoint (AWS CLI)

Use the modify-client-vpn-endpoint command.

Modify existing banner text on a Client VPN endpoint

Use the following steps to modify existing text on a client login banner.

Modify existing banner text on a Client VPN endpoint (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint that you want to modify, choose **Actions**, and then choose **Modify Client VPN endpoint**.
4. For **Enable client login banner?**, verify that it's turned on.
5. For **Client login banner text**, replace the existing text with new text that you want displayed in a banner on AWS provided clients when a VPN session is established. Use UTF-8 encoded characters only, with a maximum of 1400 characters.
6. Choose **Modify Client VPN endpoint**.

Modify client login banner on a Client VPN endpoint (AWS CLI)

Use the modify-client-vpn-endpoint command.

View currently configured login banner

Use the following steps to view a currently configured login banner.

View current login banner for a Client VPN endpoint (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint that you want to view.
4. Verify that the **Details** tab is selected.

5. View the currently configured login banner text next to **Client login banner text**.

View currently configured login banner for a Client VPN endpoint (AWS CLI)

Use the describe-client-vpn-endpoints command.

Client VPN endpoints

All client VPN sessions terminate at the Client VPN endpoint. You configure the Client VPN endpoint to manage and control all client VPN sessions.

Contents

Create a Client VPN endpoint

Create a Client VPN endpoint to enable your clients to establish a VPN session.

The Client VPN must be created in the same AWS account in which the intended target network is provisioned.

Prerequisites

Before you begin, ensure that you do the following:

- Review the rules and limitations in Limitations and rules of Client VPN (p. 3).
- Generate the server certificate, and if required, the client certificate. For more information, see Client authentication (p. 6).

To create a Client VPN endpoint (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints** and then choose **Create Client VPN Endpoint**.
3. (Optional) Provide a name tag and description for the Client VPN endpoint.
4. For **Client IPv4 CIDR**, specify an IP address range, in CIDR notation, from which to assign client IP addresses. For example, `10.0.0.0/22`.

   Note
   The address range cannot overlap with the target network address range, the VPC address range or any of the routes that will be associated with the Client VPN endpoint. The client address range must be at minimum /22 and not greater than /12 CIDR block size. You cannot change the client address range after you create the Client VPN endpoint.
5. For **Server certificate ARN**, specify the ARN for the TLS certificate to be used by the server. Clients use the server certificate to authenticate the Client VPN endpoint to which they are connecting.

   Note
   The server certificate must be present in AWS Certificate Manager (ACM) in the region you are creating the Client VPN endpoint. The certificate can either be provisioned with ACM or imported into ACM.

6. Specify the authentication method to be used to authenticate clients when they establish a VPN connection. You must select an authentication method.

   - To use user-based authentication, select **Use user-based authentication**, and then choose one of the following:
     - **Active Directory authentication**: Choose this option for Active Directory authentication. For **Directory ID**, specify the ID of the Active Directory to use.
     - **Federated authentication**: Choose this option for SAML-based federated authentication.

       For **SAML provider ARN**, specify the ARN of the IAM SAML identity provider.

       (Optional) For **Self-service SAML provider ARN**, specify the ARN of the IAM SAML identity provider that you created to support the self-service portal (p. 13), if applicable.
   - To use mutual certificate authentication, select **Use mutual authentication**, and then for **Client certificate ARN**, specify the ARN of the client certificate that's provisioned in AWS Certificate Manager (ACM).

     Note
     If the server and client certificates have been issued by the same Certificate Authority (CA), you can use the server certificate ARN for both server and client. If the client certificate was issued by a different CA, then the client certificate ARN should be specified.
7. (Optional) For **Connection logging**, specify whether to log data about client connections using Amazon CloudWatch Logs. Turn on **Enable log details on client connections**. For **CloudWatch Logs log group name**, enter the name of the log group to use. For **CloudWatch Logs log stream name**, enter the name of the log stream to use, or leave this option blank to let us create a log stream for you.
8. (Optional) For **Client Connect Handler**, turn on **Enable client connect handler** to run custom code that allows or denies a new connection to the Client VPN endpoint. For **Client Connect Handler ARN**, specify the Amazon Resource Name (ARN) of the Lambda function that contains the logic that allows or denies connections.
9. (Optional) Specify which DNS servers to use for DNS resolution. To use custom DNS servers, for **DNS Server 1 IP address** and **DNS Server 2 IP address**, specify the IP addresses of the DNS servers to use. To use VPC DNS server, for either **DNS Server 1 IP address** or **DNS Server 2 IP address**, specify the IP addresses, and add the VPC DNS server IP address.

   Note
   Verify that the DNS servers can be reached by clients.
10. (Optional) By default, the Client VPN endpoint uses the `UDP` transport protocol. To use the `TCP` transport protocol instead, for **Transport Protocol**, select **TCP**.

    Note
    UDP typically offers better performance than TCP. You cannot change the transport protocol after you create the Client VPN endpoint.
11. (Optional) To have the endpoint be a split-tunnel Client VPN endpoint, turn on **Enable split-tunnel**. By default, split-tunnel on a Client VPN endpoint is disabled.
12. (Optional) For **VPC ID**, choose the VPC to associate with the Client VPN endpoint. For **Security Group IDs**, choose one or more of the VPC's security groups to apply to the Client VPN endpoint.
13. (Optional) For **VPN port**, choose the VPN port number. The default is 443.
14. (Optional) To generate a self-service portal URL (p. 41) for clients, turn on **Enable self-service portal**.
15. (Optional) For **Session timeout hours**, choose the desired maximum VPN session duration time in hours from the available options, or leave set to default of 24 hours.
16. (Optional) Specify whether to enable client login banner text. Turn on **Enable client login banner**. For **Client login banner text**, enter the text that will be displayed in a banner on AWS provided

clients when a VPN session is established. UTF-8 encoded characters only. Maximum of 1400 characters.

17. Choose **Create Client VPN endpoint**.

After you create the Client VPN endpoint, do the following to complete the configuration and enable clients to connect:

- The initial state of the Client VPN endpoint is `pending-associate`. Clients can only connect to the Client VPN endpoint after you associate the first target network (p. 59).
- Create an authorization rule (p. 42) to specify which clients have access to the network.
- Download and prepare the Client VPN endpoint configuration file (p. 55) to distribute to your clients.
- Instruct your clients to use the AWS provided client or another OpenVPN-based client application to connect to the Client VPN endpoint. For more information, see the AWS Client VPN User Guide.

To create a Client VPN endpoint (AWS CLI)

Use the create-client-vpn-endpoint command.

Modify a Client VPN endpoint

After a Client VPN has been created, you can modify any of the following settings:

- The description
- The server certificate
- The client connection logging options
- The client connect handler option
- The DNS servers
- The split-tunnel option
- The VPC and security group associations
- The VPN port number
- The self-service portal option
- The maximum VPN session duration
- Enable or disable client login banner text
- Client login banner text

You cannot modify the client IPv4 CIDR range, authentication options, or transport protocol after the Client VPN endpoint has been created.

When you modify any of the following parameters on a Client VPN endpoint, the connection resets:

- The server certificate
- The DNS servers
- The split-tunnel option (turning support on or off)
- Routes (when you use the split-tunnel option)
- Certificate Revocation List (CRL)
- Authorization rules
- The VPN port number

You can modify a Client VPN endpoint by using the console or the AWS CLI.

To modify a Client VPN endpoint (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint to modify, choose **Actions**, and then choose **Modify Client VPN endpoint**.
4. For **Description**, enter a brief description for the Client VPN endpoint.
5. For **Server certificate ARN**, specify the ARN for the TLS certificate to be used by the server. Clients use the server certificate to authenticate the Client VPN endpoint to which they are connecting.

   Note
   The server certificate must be present in AWS Certificate Manager (ACM) in the region you are creating the Client VPN endpoint. The certificate can either be provisioned with ACM or imported into ACM.
6. Specify whether to log data about client connections using Amazon CloudWatch Logs. For **Enable log details on client connections**, do one of the following:

   - To activate client connection logging, turn on **Enable log details on client connections**. For **CloudWatch Logs log group name**, select the name of the log group to use. For **CloudWatch Logs log stream name**, select the name of the log stream to use, or leave this option blank to let us create a log stream for you.
   - To deactivate client connection logging, turn off **Enable log details on client connections**.
7. For **Client connect handler**, to activate the client connect handler (p. 14) turn on **Enable client connect handler**. For **Client Connect Handler ARN**, specify the Amazon Resource Name (ARN) of the Lambda function that contains the logic that allows or denies connections.
8. Turn on or off **Enable DNS servers**. To use custom DNS servers, for **DNS Server 1 IP address** and **DNS Server 2 IP address**, specify the IP addresses of the DNS servers to use. To use VPC DNS server, for either **DNS Server 1 IP address** or **DNS Server 2 IP address**, specify the IP addresses, and add the VPC DNS server IP address.

   Note
   Verify that the DNS servers can be reached by clients.
9. Turn on or off **Enable split-tunnel**. By default, split-tunnel on a VPN endpoint is off.
10. For **VPC ID**, choose the VPC to associate with the Client VPN endpoint. For **Security Group IDs**, choose one or more of the VPC's security groups to apply to the Client VPN endpoint.
11. For **VPN port**, choose the VPN port number. The default is 443.
12. To generate a self-service portal URL (p. 41) for clients, turn on **Enable self-service portal**.
13. For **Session timeout hours**, choose the desired maximum VPN session duration time in hours from the available options, or leave set to default of 24 hours.
14. Turn on or off **Enable client login banner**. If you want to use the client login banner, enter the text that will be displayed in a banner on AWS provided clients when a VPN session is established. UTF-8 encoded characters only. Maximum of 1400 characters.
15. Choose **Modify Client VPN endpoint**.

To modify a Client VPN endpoint (AWS CLI)

Use the modify-client-vpn-endpoint command.

View Client VPN endpoints

You can view information about Client VPN endpoints by using the console or the AWS CLI.

To view Client VPN endpoints (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint to view.
4. Use the **Details**, **Target network associations**, **Security groups**, **Authorization rules**, **Route table**, **Connections** and **Tags** tabs to view information about existing Client VPN endpoints.

   You can also use filters to help refine your search.

To view Client VPN endpoints (AWS CLI)

Use the describe-client-vpn-endpoints command.

Delete a Client VPN endpoint

You will need to disassociate all target networks before you can delete a Client VPN endpoint. When you delete a Client VPN endpoint, its state is changed to `deleting` and clients can no longer connect to it.

You can delete a Client VPN endpoint by using the console or the AWS CLI.

To delete a Client VPN endpoint (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint to delete. Choose **Actions**, **Delete Client VPN endpoint**.
4. Enter *delete* into the confirmation window and choose **Delete**.

To delete a Client VPN endpoint (AWS CLI)

Use the delete-client-vpn-endpoint command.

Working with connection logs

You can enable connection logging for a new or existing Client VPN endpoint, and start capturing connection logs.

Before you begin, you must have a CloudWatch Logs log group in your account. For more information, see Working with Log Groups and Log Streams in the *Amazon CloudWatch Logs User Guide*. Charges apply for using CloudWatch Logs. For more information, see Amazon CloudWatch pricing.

When you enable connection logging, you can specify the name of a log stream in the log group. If you do not specify a log stream, the Client VPN service creates one for you.

Enable connection logging for a new Client VPN endpoint

You can enable connection logging when you create a new Client VPN endpoint by using the console or the command line.

To enable connection logging for a new Client VPN endpoint using the console

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

2. In the navigation pane, choose **Client VPN Endpoints**, and then choose **Create Client VPN endpoint.**
3. Complete the options until you reach the **Connection Logging** section. For more information about the options, see Create a Client VPN endpoint (p. 49).
4. Under **Connection logging**, turn on **Enable log details on client connections**.
5. For **CloudWatch Logs log group name**, choose the name of the CloudWatch Logs log group.
6. (Optional) For **CloudWatch Logs log stream name**, choose the name of the CloudWatch Logs log stream.
7. Choose **Create Client VPN endpoint**.

To enable connection logging for a new Client VPN endpoint using the AWS CLI

Use the create-client-vpn-endpoint command, and specify the `--connection-log-options` parameter. You can specify the connection logs information in JSON format, as shown in the following example.

```
{
    "Enabled": true,
    "CloudwatchLogGroup": "ClientVpnConnectionLogs",
    "CloudwatchLogStream": "NewYorkOfficeVPN"
}
```

Enable connection logging for an existing Client VPN endpoint

You can enable connection logging for an existing Client VPN endpoint by using the console or the command line.

To enable connection logging for an existing Client VPN endpoint using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint, choose **Actions**, and then choose **Modify Client VPN endpoint**.
4. Under **Connection logging**, turn on **Enable log details on client connections**.
5. For **CloudWatch Logs log group name**, choose the name of the CloudWatch Logs log group.
6. (Optional) For **CloudWatch Logs log stream name**, choose the name of the CloudWatch Logs log stream.
7. Choose **Modify Client VPN endpoint**.

To enable connection logging for an existing Client VPN endpoint using the AWS CLI

Use the modify-client-vpn-endpoint command and specify the `--connection-log-options` parameter. You can specify the connection logs information in JSON format, as shown in the following example.

```
{
    "Enabled": true,
    "CloudwatchLogGroup": "ClientVpnConnectionLogs",
    "CloudwatchLogStream": "NewYorkOfficeVPN"
}
```

View connection logs

You can view your connection logs using the CloudWatch Logs console.

To view your connection logs using the console

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Log groups**, and select the log group that contains your connection logs.
3. Select the log stream for your Client VPN endpoint.

   Note
   The **Timestamp** column displays the time that the connection log was published to CloudWatch Logs, not the time of the connection.

For more information about searching log data, see Search Log Data Using Filter Patterns in the *Amazon CloudWatch Logs User Guide*.

Turn off connection logging

You can turn off connection logging for a Client VPN endpoint by using the console or the command line. When you turn off connection logging, existing connection logs in CloudWatch Logs are not deleted.

To turn off connection logging using the console

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint, choose **Actions**, and then choose **Modify Client VPN endpoint**.
4. Under **Connection logging**, turn off **Enable log details on client connections**.
5. Choose **Modify Client VPN endpoint**.

To turn off connection logging using the AWS CLI

Use the modify-client-vpn-endpoint command, and specify the `--connection-log-options` parameter. Ensure that `Enabled` is set to `false`.

Export and configure the client configuration file

The Client VPN endpoint configuration file is the file that clients (users) use to establish a VPN connection with the Client VPN endpoint. You must download (export) this file and distribute it to all clients who need access to the VPN. Alternatively, if you've enabled the self-service portal for your Client VPN endpoint, clients can log into the portal and download the configuration file themselves. For more information, see Access the self-service portal (p. 41).

If your Client VPN endpoint uses mutual authentication, you must add the client certificate and the client private key to the .ovpn configuration file (p. 56) that you download. After you add the information, clients can import the .ovpn file into the OpenVPN client software.

> **Important**
> If you do not add the client certificate and the client private key information to the file, clients that authenticate using mutual authentication cannot connect to the Client VPN endpoint.

By default, the “--remote-random-hostname” option in the OpenVPN client configuration enables wildcard DNS. Because wildcard DNS is enabled, the client does not cache the IP address of the endpoint and you will not be able to ping the DNS name of the endpoint.

If your Client VPN endpoint uses Active Directory authentication and if you enable multi-factor authentication (MFA) on your directory after you distribute the client configuration file, you must download a new file and redistribute it to your clients. Clients cannot use the previous configuration file to connect to the Client VPN endpoint.

Export the client configuration file

You can export the client configuration by using the console or the AWS CLI.

To export client configuration (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint for which to download the client configuration and choose **Download Client Configuration**.

To export client configuration (AWS CLI)

Use the export-client-vpn-client-configuration command and specify the output file name.

```
$ aws ec2 export-client-vpn-client-configuration --client-vpn-endpoint-id endpoint_id --
output text>config_filename.ovpn
```

Add the client certificate and key information (mutual authentication)

If your Client VPN endpoint uses mutual authentication, you must add the client certificate and the client private key to the .ovpn configuration file that you download.

You cannot modify the client certificate when you use mutual authentication.

To add the client certificate and key information (mutual authentication)

You can use one of the following options.

(Option 1) Distribute the client certificate and key to clients along with the Client VPN endpoint configuration file. In this case, specify the path to the certificate and key in the configuration file. Open the configuration file using your preferred text editor, and add the following to the end of the file. Replace */path/* with the location of the client certificate and key (the location is relative to the client that's connecting to the endpoint).

```
cert /path/client1.domain.tld.crt
key /path/client1.domain.tld.key
```

(Option 2) Add the contents of the client certificate between `<cert></cert>` tags and the contents of the private key between `<key></key>` tags to the configuration file. If you choose this option, you distribute only the configuration file to your clients.

If you generated separate client certificates and keys for each user that will connect to the Client VPN endpoint, repeat this step for each user.

The following is an example of the format of a Client VPN configuration file that includes the client certificate and key.

```
client
dev tun
proto udp
remote asdf.cvpn-endpoint-0011abcabcabcabc1.prod.clientvpn.eu-west-2.amazonaws.com 443
remote-random-hostname
resolv-retry infinite
nobind
remote-cert-tls server
cipher AES-256-GCM
verb 3

<ca>
Contents of CA
</ca>

<cert>
Contents of client certificate (.crt) file
</cert>

<key>
Contents of private key (.key) file
</key>

reneg-sec 0
```

Routes

Each Client VPN endpoint has a route table that describes the available destination network routes. Each route in the route table determines where the network traffic is directed. You must configure authorization rules for each Client VPN endpoint route to specify which clients have access to the destination network.

When you associate a subnet from a VPC with a Client VPN endpoint, a route for the VPC is automatically added to the Client VPN endpoint's route table. To enable access for additional networks, such as peered VPCs, on-premises networks, the local network (to enable clients to communicate with each other), or the internet, you must manually add a route to the Client VPN endpoint's route table.

> **Note**
> If you are associating multiple subnets to the Client VPN endpoint, you should make sure to create a route for each subnet as described here Access to a peered VPC, Amazon S3, or the internet is intermittent (p. 80). Each associated subnet should have an identical set of routes.

Contents

Split-tunnel on Client VPN endpoint considerations

When you use split-tunnel on a Client VPN endpoint, all of the routes that are in the Client VPN route tables are added to the client route table when the VPN is established. If you add a route after the VPN is established, you must reset the connection so that the new route is sent to the client.

We recommend that you account for the number of routes that the client device can handle before you modify the Client VPN endpoint route table.

Create an endpoint route

When you create a route, you specify how traffic for the destination network should be directed.

To allow clients to access the internet, add a destination `0.0.0.0/0` route.

You can add routes to a Client VPN endpoint by using the console and the AWS CLI.

To create a Client VPN endpoint route (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint to which to add the route, choose **Route table**, and then choose **Create route**.
4. For **Route destination**, specify the IPv4 CIDR range for the destination network. For example:
   - To add a route for the VPC of the Client VPN endpoint, enter the VPC's IPv4 CIDR range.
   - To add a route for internet access, enter `0.0.0.0/0`.
   - To add a route for a peered VPC, enter the peered VPC's IPv4 CIDR range.
   - To add a route for an on-premises network, enter the AWS Site-to-Site VPN connection's IPv4 CIDR range.
5. For **Subnet ID for target network association**, select the subnet that is associated with the Client VPN endpoint.

   Alternatively, if you're adding a route for the local Client VPN endpoint network, select `local`.
6. (Optional) For **Description**, enter a brief description for the route.
7. Choose **Create route**.

To create a Client VPN endpoint route (AWS CLI)

Use the create-client-vpn-route command.

View endpoint routes

You can view the routes for a specific Client VPN endpoint by using the console or the AWS CLI.

To view Client VPN endpoint routes (console)

1. In the navigation pane, choose **Client VPN Endpoints**.
2. Select the Client VPN endpoint for which to view routes and choose **Route table**.

To view Client VPN endpoint routes (AWS CLI)

Use the describe-client-vpn-routes command.

Delete an endpoint route

You can only delete routes that you added manually. You can't delete routes that were automatically added when you associated a subnet with the Client VPN endpoint. To delete routes that were automatically added, you must disassociate the subnet that initiated its creation from the Client VPN endpoint.

You can delete a route from a Client VPN endpoint by using the console or the AWS CLI.

To delete a Client VPN endpoint route (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint from which to delete the route and choose **Route table**.
4. Select the route to delete, choose **Delete route**, and choose **Delete route**.

To delete a Client VPN endpoint route (AWS CLI)

Use the delete-client-vpn-route command.

Target networks

A target network is a subnet in a VPC. A Client VPN endpoint must have at least one target network to enable clients to connect to it and establish a VPN connection.

For more information about the kinds of access you can configure (such as enabling your clients to access the internet), see Scenarios and examples (p. 22).

Contents

Associate a target network with a Client VPN endpoint

You can associate one or more target networks (subnets) with a Client VPN endpoint.

The following rules apply:

- The subnet must have a CIDR block with at least a /27 bitmask, for example 10.0.0.0/27. The subnet must also have at least 8 available IP addresses.
- The subnet's CIDR block cannot overlap with the client CIDR range of the Client VPN endpoint.
- If you associate more than one subnet with a Client VPN endpoint, each subnet must be in a different Availability Zone. We recommend that you associate at least two subnets to provide Availability Zone redundancy.
- If you specified a VPC when you created the Client VPN endpoint, the subnet must be in the same VPC. If you haven't yet associated a VPC with the Client VPN endpoint, you can choose any subnet in any VPC.

  All further subnet associations must be from the same VPC. To associate a subnet from a different VPC, you must first modify the Client VPN endpoint and change the VPC that's associated with it. For more information, see Modify a Client VPN endpoint (p. 51).

When you associate a subnet with a Client VPN endpoint, we automatically add the local route of the VPC in which the associated subnet is provisioned to the Client VPN endpoint's route table.

> **Note**
> After your target networks are associated, when you add or remove additional CIDRs to your attached VPC, you must perform one of the following operations to update the local route for your Client VPN endpoint route table:
>
> - Disassociate your Client VPN endpoint from the target network, and then associate the Client VPN endpoint to the target network.
> - Manually add the route to, or remove the route from the Client VPN endpoint route table.

After you associate the first subnet with the Client VPN endpoint, the Client VPN endpoint's status changes from `pending-associate` to `available` and clients are able to establish a VPN connection.

To associate a target network with a Client VPN endpoint (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint with which to associate the target network, choose **Target network associations**, and then choose **Associate target network**.
4. For **VPC**, choose the VPC in which the subnet is located. If you specified a VPC when you created the Client VPN endpoint or if you have previous subnet associations, it must be the same VPC.
5. For **Choose a subnet to associate**, choose the subnet to associate with the Client VPN endpoint.
6. Choose **Associate target network**.

To associate a target network with a Client VPN endpoint (AWS CLI)

Use the associate-client-vpn-target-network command.

Apply a security group to a target network

When you create a Client VPN endpoint, you can specify the security groups to apply to the target network. When you associate the first target network with a Client VPN endpoint, we automatically apply the default security group of the VPC in which the associated subnet is located. For more information, see Security groups (p. 13).

You can change the security groups for the Client VPN endpoint. The security group rules that you require depend on the kind of VPN access you want to configure. For more information, see Scenarios and examples (p. 22).

To apply a security group to a target network (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint to which to apply the security groups.
4. Choose **Security Groups**, and then choose **Apply Security Groups**.
5. Select the appropriate security group(s) from **Security group IDs**.
6. Choose **Apply Security Groups**.

To apply a security group to a target network (AWS CLI)

Use the apply-security-groups-to-client-vpn-target-network command.

Disassociate a target network from a Client VPN endpoint

If you disassociate all target networks from a Client VPN endpoint, clients can no longer establish a VPN connection. When you disassociate a subnet, we remove the route that was automatically created when the association was made.

To disassociate a target network from a Client VPN endpoint (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint with which the target network is associated and choose **Target network associations**.
4. Select the target network to disassociate, choose **Disassociate**, and then choose **Disassociate target network**.

To disassociate a target network from a Client VPN endpoint (AWS CLI)

Use the disassociate-client-vpn-target-network command.

View target networks

You can view the targets associated with a Client VPN endpoint using the console or the AWS CLI.

To view target networks (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the appropriate Client VPN endpoint and choose **Target network associations**.

To view target networks using the AWS CLI

Use the describe-client-vpn-target-networks command.

VPN session maximum duration

AWS Client VPN provides several options for the maximum VPN session duration. You can configure a shorter maximum VPN session duration to meet security and compliance requirements. By default, the maximum VPN session duration is 24 hours.

> **Note**
> When the maximum VPN session duration value is decreased, active VPN sessions older than the new timeout value will be disconnected.

See Release notes for the AWS provided client in the *AWS Client VPN User Guide* for details on client desktop applications.

Contents

Configure maximum VPN session during creation of a Client VPN endpoint

For detailed steps for configuring maximum VPN session during creation of a Client VPN endpoint, see Create a Client VPN endpoint (p. 49).

View current maximum VPN session duration

Use the following steps to view current maximum VPN session duration.

View current maximum VPN session duration for a Client VPN endpoint (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN Endpoints**.
3. Select the Client VPN endpoint that you want to view.
4. Verify that the **Details** tab is selected.
5. View the current maximum VPN session duration next to **Session timeout hours**.

View current maximum VPN session duration for a Client VPN endpoint (AWS CLI)

Use the describe-client-vpn-endpoints command.

Modify maximum VPN session duration

Use the following steps to modify an existing maximum VPN session duration.

Modify an existing maximum VPN session duration for a Client VPN endpoint (console)

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Client VPN endpoints**.
3. Select the Client VPN endpoint that you want to modify, choose **Actions**, and then choose **Modify Client VPN Endpoint**.
4. For **Session timeout hours**, choose the desired maximum VPN session duration time in hours.
5. Choose **Modify Client VPN endpoint**.

Modify an existing maximum VPN session duration for a Client VPN endpoint (AWS CLI)

Use the modify-client-vpn-endpoint command.

Security in AWS Client VPN

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to AWS Client VPN, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Client VPN. The following topics show you how to configure Client VPN to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Client VPN resources.

Contents

Data protection in AWS Client VPN

The AWS shared responsibility model applies to data protection in AWS Client VPN. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.

- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a **Name** field. This includes when you work with Client VPN or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Encryption in transit

AWS Client VPN provides secure connections from any location using Transport Layer Security (TLS) 1.2 or later.

Internetwork traffic privacy

Enabling internetwork access

You can enable clients to connect to your VPC and other networks through a Client VPN endpoint. For more information and examples, see Scenarios and examples (p. 22).

Restricting access to networks

You can configure your Client VPN endpoint to restrict access to specific resources in your VPC. For user-based authentication, you can also restrict access to parts of your network, based on the user group that accesses the Client VPN endpoint. For more information, see Restrict access to your network (p. 32).

Authenticating clients

Authentication is implemented at the first point of entry into the AWS Cloud. It is used to determine whether clients are allowed to connect to the Client VPN endpoint. If authentication succeeds, clients connect to the Client VPN endpoint and establish a VPN session. If authentication fails, the connection is denied and the client is prevented from establishing a VPN session.

Client VPN offers the following types of client authentication:

- Active Directory authentication (p. 6) (user-based)
- Mutual authentication (p. 6) (certificate-based)
- Single sign-on (SAML-based federated authentication) (p. 9) (user-based)

Identity and access management for Client VPN

AWS uses security credentials to identify you and to grant you access to your AWS resources. You can use features of AWS Identity and Access Management (IAM) to allow other users, services, and applications to use your AWS resources fully or in a limited way, without sharing your security credentials.

By default, IAM users don't have permission to create, view, or modify AWS resources. To allow an IAM user to access resources, such as a Client VPN endpoint, and perform tasks, you must create an IAM policy. This policy must grant the IAM user permission to use the specific resources and API actions they need. Then, attach the policy to the IAM user or the group to which the IAM user belongs. When you attach a policy to a user or group of users, it allows or denies the users permission to perform the specified tasks on the specified resources.

For example, the following policy enables read-only access. Users can view Client VPN endpoints and their components, but they cannot create, modify, or delete them.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeClientVpnRoutes",
                "ec2:DescribeClientVpnAuthorizationRules",
                "ec2:DescribeClientVpnConnections",
                "ec2:DescribeClientVpnTargetNetworks",
                "ec2:DescribeClientVpnEndpoints"
            ],
            "Resource": "*"
        }
    ]
}
```

You can also use resource-level permissions to restrict what resources users can use when they invoke Client VPN actions. For example, the following policy allows users to work with Client VPN endpoints, but only if the Client VPN endpoint has the tag `purpose=test`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteClientVpnEndpoint",
                "ec2:ModifyClientVpnEndpoint",
                "ec2:AssociateClientVpnTargetNetwork",
                "ec2:DisassociateClientVpnTargetNetwork",
                "ec2:ApplySecurityGroupsToClientVpnTargetNetwork",
                "ec2:AuthorizeClientVpnIngress",
                "ec2:CreateClientVpnRoute",
                "ec2:DeleteClientVpnRoute",
                "ec2:RevokeClientVpnIngress"
            ],
            "Resource": "arn:aws:ec2:*:*:client-vpn-endpoint/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/purpose": "test"
                }
            }
        }
    ]
}
```

For more information about IAM, see the IAM User Guide. For a list of Amazon EC2 actions, including Client VPN actions, see Actions, Resources, and Condition Keys for Amazon EC2 in the *IAM User Guide*.

Using service-linked roles for Client VPN

AWS Client VPN uses service-linked roles for the permissions that it requires to call other AWS services on your behalf. For more information, see Using Service-Linked Roles in the *IAM User Guide*.

Service-linked role permissions for Client VPN

AWS Client VPN uses the service-linked role named **AWSServiceRoleForClientVPN** to call the following actions on your behalf when you work with Client VPN endpoints:

- `ec2:CreateNetworkInterface`
- `ec2:CreateNetworkInterfacePermission`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeVpcs`
- `ec2:DescribeSubnets`
- `ec2:DescribeInternetGateways`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:DeleteNetworkInterface`
- `ec2:DescribeAccountAttributes`
- `ds:AuthorizeApplication`
- `ds:DescribeDirectories`
- `ds:GetDirectoryLimits`
- `ds:ListAuthorizedApplications`
- `ds:UnauthorizeApplication`
- `lambda:GetFunctionConfiguration`
- `logs:DescribeLogStreams`
- `logs:CreateLogStream`
- `logs:PutLogEvents`
- `logs:DescribeLogGroups`
- `acm:GetCertificate`
- `acm:DescribeCertificate`

The **AWSServiceRoleForClientVPN** service-linked role trusts the clientvpn.amazonaws.com principal to assume the role.

If you use the client connect handler for your Client VPN endpoint, Client VPN uses a service-linked role called **AWSServiceRoleForClientVPNConnections**. This role gets permissions from the **ClientVPNServiceConnectionsRolePolicy** policy that allows Client VPN to invoke Lambda functions on your behalf. The policy allows the `lambda:InvokeFunction` action only on Lambda functions with the `AWSClientVPN-` prefix. For more information, see Connection authorization (p. 14).

Creating service-linked roles for Client VPN

You don't need to manually create the **AWSServiceRoleForClientVPN** or the **AWSServiceRoleForClientVPNConnections** roles. Client VPN creates the roles for you when you create the first Client VPN endpoint in your account.

For Client VPN to create the service-linked roles on your behalf, you must have the required permissions. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Editing a service-linked role for Client VPN

You cannot edit the **AWSServiceRoleForClientVPN** or **AWSServiceRoleForClientVPNConnections** service-linked roles.

Deleting a service-linked role for Client VPN

If you no longer need to use Client VPN, we recommend that you delete the **AWSServiceRoleForClientVPN** and **AWSServiceRoleForClientVPNConnections** service-linked roles.

You must first delete the related Client VPN resources. This ensures that you do not inadvertently remove permission to access the resources.

Use the IAM console, the IAM CLI, or the IAM API to delete the service-linked roles. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Logging and monitoring

Monitoring is an important part of maintaining the reliability, availability, and performance of your Client VPN endpoint. You should collect monitoring data from all of the parts of your solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your resources and responding to potential incidents.

Amazon CloudWatch

Amazon CloudWatch monitors your AWS resources and the applications that you run on AWS in real time. You can collect and track metrics for your Client VPN endpoint. For more information, see Monitoring with Amazon CloudWatch (p. 70).

AWS CloudTrail

AWS CloudTrail captures Amazon EC2 API calls and related events made by or on behalf of your AWS account. It then delivers the log files to an Amazon S3 bucket that you specify. For more information, see Monitoring with AWS CloudTrail (p. 72).

Amazon CloudWatch Logs

You can view connection logs to get information about connection events, which are when clients connect, attempt to connect, or disconnect from your Client VPN endpoint. For more information, see Connection logging (p. 19).

Resilience in AWS Client VPN

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

In addition to the AWS global infrastructure, AWS Client VPN offers features to help support your data resiliency and backup needs.

Multiple target networks for high availability

You associate a target network with a Client VPN endpoint to enable clients to establish VPN sessions. Target networks are subnets in your VPC. Each subnet that you associate with the Client VPN endpoint must belong to a different Availability Zone. You can associate multiple subnets with a Client VPN endpoint for high availability.

Infrastructure security in AWS Client VPN

As a managed service, AWS Client VPN is protected by the AWS global network security procedures that are described in the Security Pillar of the AWS Well-Architected Framework.

You use AWS published API calls to access Client VPN through the network. Clients must support Transport Layer Security (TLS) 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Security best practices for AWS Client VPN

AWS Client VPN provides a number of security features to consider as you develop and implement your own security policies. The following best practices are general guidelines and don't represent a complete security solution. Because these best practices might not be appropriate or sufficient for your environment, treat them as helpful considerations rather than prescriptions.

Authorization rules

Use authorization rules to restrict which users can access your network. For more information, see Authorization rules (p. 42).

Security groups

Use security groups to control which resources users can access in your VPC. For more information, see Security groups (p. 13).

Client certificate revocation lists

Use client certificate revocation lists to revoke access to a Client VPN endpoint for specific client certificates. For example, when a user leaves your organization. For more information, see Client certificate revocation lists (p. 43).

Monitoring tools

Use monitoring tools to keep track of availability and performance of your Client VPN endpoints. For more information, see Monitoring Client VPN (p. 70).

Identity and access management

Manage access to Client VPN resources and APIs by using IAM policies for your IAM users and IAM roles. For more information, see Identity and access management for Client VPN (p. 64).

IPv6 Considerations

Currently the Client VPN service does not support routing IPv6 traffic through the VPN tunnel. However, there are cases when IPv6 traffic should be routed into the VPN tunnel to prevent IPv6 leak. IPv6 leak can happen when both IPv4 and IPv6 are enabled and connected to the VPN, but the VPN doesn't route IPv6 traffic into its tunnel. In this case, when connecting to an IPv6 enabled destination, you are actually still connecting with your IPv6 address provided by your ISP. This will leak your real IPv6 address. The instructions below explain how to route IPv6 traffic into the VPN tunnel.

The following IPv6-related directives should be added to your Client VPN configuration file to prevent IPv6 leak:

```
ifconfig-ipv6 arg0 arg1
route-ipv6 arg0
```

An example might be:

```
ifconfig-ipv6 fd15:53b6:dead::2 fd15:53b6:dead::1
route-ipv6 2000::/4
```

In this example, `ifconfig-ipv6 fd15:53b6:dead::2 fd15:53b6:dead::1` will set the local tunnel device IPv6 address to be `fd15:53b6:dead::2` and the remote VPN endpoint IPv6 address to be `fd15:53b6:dead::1`.

The next command, `route-ipv6 2000::/4` will route IPv6 addresses from `2000:0000:0000:0000:0000:0000:0000:0000` to `2fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff` into the VPN connection.

> **Note**
> For "TAP" device routing in Windows for example, the second parameter of `ifconfig-ipv6` will be used as route target for `--route-ipv6`.

Organizations should configure the two parameters of `ifconfig-ipv6` themselves, and can use addresses in `100::/64` (from `0100:0000:0000:0000:0000:0000:0000:0000` to `0100:0000:0000:0000:ffff:ffff:ffff:ffff`) or `fc00::/7` (from `fc00:0000:0000:0000:0000:0000:0000:0000` to `fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff`). `100::/64` is Discard-Only Address Block, and `fc00::/7` is Unique-Local.

Another example:

```
ifconfig-ipv6 fd15:53b6:dead::2 fd15:53b6:dead::1
route-ipv6 2000::/3
route-ipv6 fc00::/7
```

In this example, the configuration will route all currently allocated IPv6 traffic into the VPN connection.

Verification

Your organization will likely have its own tests. A basic verification is to set up a full tunnel VPN connection, then run ping6 to an IPv6 server using the IPv6 address. The IPv6 address of the server should be in the range specified by the `route-ipv6` command. This ping test should fail. However, this may change if IPv6 support is added to the Client VPN service in the future. If the ping is successful and you are able to access public sites when connected in full tunnel mode, you may need to do further troubleshooting. You can also test by using some publicly available tools like ipleak.org as well.

Monitoring Client VPN

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Client VPN and your other AWS solutions. You can use the following features to monitor your Client VPN endpoints, analyze traffic patterns, and troubleshoot issues with your Client VPN endpoints.

Amazon CloudWatch

Monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the Amazon CloudWatch User Guide.

AWS CloudTrail

Captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

Amazon CloudWatch Logs

Enables you to monitor connection attempts made to your AWS Client VPN endpoint. You can view the connection attempts and connection resets for the Client VPN connections. For the connection attempts, you can see both the successful and failed connection attempts. You can specify the CloudWatch Logs log stream to log the connection details. For more information, see Connection logging (p. 19) and the Amazon CloudWatch Logs User Guide.

Monitoring with Amazon CloudWatch

AWS Client VPN publishes the following metrics to Amazon CloudWatch for your Client VPN endpoints. Metrics are published to Amazon CloudWatch every five minutes.

Metric	Description
ActiveConnectionsCount	The number of active connections to the Client VPN endpoint. Units: Count
AuthenticationFailures	The number of authentication failures for the Client VPN endpoint. Units: Count
CrlDaysToExpiry	The number of days until the Certificate Revocation List (CRL) which is configured on the Client VPN endpoint expires. Units: Days
EgressBytes	The number of bytes sent from the Client VPN endpoint.

Metric	Description
	Units: Bytes
EgressPackets	The number of packets sent from the Client VPN endpoint. Units: Count
IngressBytes	The number of bytes received by the Client VPN endpoint. Units: Bytes
IngressPackets	The number of packets received by the Client VPN endpoint. Units: Count
SelfServicePortalClientConfigurationDownloads	The number of downloads of the Client VPN endpoint configuration file from the self-service portal. Unit: Count

AWS Client VPN publishes the following posture assessment (p. 16) metrics for your Client VPN endpoints.

Metric	Description
ClientConnectHandlerTimeouts	The number of timeouts on invoking the client connect handler for connections to the Client VPN endpoint. Units: Count
ClientConnectHandlerInvalidResponses	The number of invalid responses returned by the client connect handler for connections to the Client VPN endpoint. Units: Count
ClientConnectHandlerOtherExecutionErrors	The number of unexpected errors while running the client connect handler for connections to the Client VPN endpoint. Units: Count
ClientConnectHandlerThrottlingErrors	The number of throttling errors on invoking the client connect handler for connections to the Client VPN endpoint. Units: Count
ClientConnectHandlerDeniedConnections	The number of connections denied by the client connect handler for connections to the Client VPN endpoint. Units: Count

Metric	Description
ClientConnectHandlerFailedServiceErrors	The number of service side errors while running the client connect handler for connections to the Client VPN endpoint. Units: Count

You can filter the metrics for your Client VPN endpoint by endpoint.

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.

For more information, see the Amazon CloudWatch User Guide.

Viewing CloudWatch metrics

You can view the metrics for your Client VPN endpoint as follows.

To view metrics using the CloudWatch console

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Under **All metrics**, choose the **ClientVPN** metric namespace.
4. To view the metrics, select the metric dimension **by endpoint**.

To view metrics using the AWS CLI

At a command prompt, use the following command to list the metrics that are available for the Client VPN

```
aws cloudwatch list-metrics --namespace "AWS/ClientVPN"
```

Monitoring with AWS CloudTrail

AWS Client VPN is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Client VPN. CloudTrail captures all API calls for Client VPN as events. The calls captured include calls from the Client VPN console and code calls to the Client VPN API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Client VPN. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Use the information collected by CloudTrail to determine the request that was made to Client VPN, the requesting IP address, the requester, when it was made, and additional details.

For more information about CloudTrail, see the AWS CloudTrail User Guide.

Client VPN information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Client VPN, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for Client VPN, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All Client VPN actions are logged by CloudTrail and are documented in the Amazon EC2 API Reference. For example, calls to the `CreateClientVpnEndpoint`, `AssociateClientVpnTargetNetwork`, and `AuthorizeClientVpnIngress` actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

Understanding Client VPN log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

For more information, see Logging Amazon EC2, Amazon EBS, and Amazon VPC API calls with AWS CloudTrail in the *Amazon EC2 API Reference*.

AWS Client VPN quotas

Your AWS account has the following quotas, formerly referred to as limits, related to Client VPN endpoints. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, and other quotas cannot be increased.

To request a quota increase for an adjustable quota, choose **Yes** in the **Adjustable** column. For more information, see Requesting a quota increase in the *Service Quotas User Guide*.

Client VPN quotas

Name	Default	Adjustable
Authorization rules per Client VPN endpoint	50	Yes
Client VPN endpoints per Region	5	Yes
Concurrent client connections per Client VPN endpoint	This value depends on the number of subnet associations per endpoint. • 1 – 7,000 • 2 – 36,500 • 3 – 66,500 • 4 – 96,500 • 5 – 126,000	Yes
Concurrent operations per Client VPN endpoint †	10	No
Entries in a client certificate revocation list for Client VPN endpoints	20,000	No
Routes per Client VPN endpoint	10	Yes

† Operations include:

- Associate or disassociate subnets
- Create or delete routes
- Create or delete inbound and outbound rules
- Create or delete security groups

Users and groups quotas

When you configure users and groups for Active Directory or a SAML-based IdP, the following quotas apply:

- Users can belong to a maximum of 200 groups. We ignore any groups after the 200th group.

- The maximum length for the group ID is 255 characters.
- The maximum length for the name ID is 255 characters. We truncate characters after the 255th character.

General considerations

Take the following into consideration when you use Client VPN endpoints:

- If you use Active Directory to authenticate the user, the Client VPN endpoint must belong to the same account as the AWS Directory Service resource used for Active Directory authentication.
- If you use SAML-based federated authentication to authenticate a user, the Client VPN endpoint must belong to the same account as the IAM SAML identity provider that you create to define the IdP-to-AWS trust relationship. The IAM SAML identity provider can be shared across multiple Client VPN endpoints in the same AWS account.

Troubleshooting Client VPN

The following topic can help you troubleshoot problems that you might have with a Client VPN endpoint.

For more information about troubleshooting OpenVPN-based software that clients use to connect to a Client VPN, see Troubleshooting Your Client VPN Connection in the *AWS Client VPN User Guide*.

Common problems

Unable to resolve Client VPN endpoint DNS name

Problem

I am unable to resolve the Client VPN endpoint's DNS name.

Cause

The Client VPN endpoint configuration file includes a parameter called `remote-random-hostname`. This parameter forces the client to prepend a random string to the DNS name to prevent DNS caching. Some clients do not recognize this parameter and therefore, they do not prepend the required random string to the DNS name.

Solution

Open the Client VPN endpoint configuration file using your preferred text editor. Locate the line that specifies the Client VPN endpoint DNS name, and prepend a random string to it so that the format is *`random_string.displayed_DNS_name`*. For example:

- Original DNS name: `cvpn-endpoint-0102bc4c2eEXAMPLE.clientvpn.us-west-2.amazonaws.com`
- Modified DNS name: `asdfa.cvpn-endpoint-0102bc4c2eEXAMPLE.clientvpn.us-west-2.amazonaws.com`

Traffic is not being split between subnets

Problem

I am trying to split network traffic between two subnets. Private traffic should be routed through a private subnet, while internet traffic should be routed through a public subnet. However, only one route is being used even though I have added both routes to the Client VPN endpoint route table.

Cause

You can associate multiple subnets with a Client VPN endpoint, but you can associate only one subnet per Availability Zone. The purpose of multiple subnet association is to provide high availability and Availability Zone redundancy for clients. However, Client VPN does not enable you to selectively split traffic between the subnets that are associated with the Client VPN endpoint.

Clients connect to a Client VPN endpoint based on the DNS round-robin algorithm. This means that their traffic can be routed through any of the associated subnets when they establish a connection. Therefore, they might experience connectivity issues if they land on an associated subnet that does not have the required route entries.

For example, say that you configure the following subnet associations and routes:

- Subnet associations
  - Association 1: Subnet-A (us-east-1a)
  - Association 2: Subnet-B (us-east-1b)
- Routes
  - Route 1: 10.0.0.0/16 routed to Subnet-A
  - Route 2: 172.31.0.0/16 routed to Subnet-B

In this example, clients that land on Subnet-A when they connect cannot access Route 2, while clients that land on Subnet-B when they connect cannot access Route 1.

Solution

Verify that the Client VPN endpoint has the same route entries with targets for each associated network. This ensures that clients have access to all routes regardless of the subnet through which their traffic is routed.

Authorization rules for Active Directory groups not working as expected

Problem

I have configured authorization rules for my Active Directory groups, but they are not working as I expected. I have added an authorization rule for `0.0.0.0/0` to authorize traffic for all networks, but traffic still fails for specific destination CIDRs.

Cause

Authorization rules are indexed on network CIDRs. Authorization rules must grant Active Directory groups access to specific network CIDRs. Authorization rules for `0.0.0.0/0` are handled as a special case, and are therefore evaluated last, regardless of the order in which the authorization rules are created.

For example, say that you create five authorization rules in the following order:

- Rule 1: Group 1 access to `10.1.0.0/16`
- Rule 2: Group 1 access to `0.0.0.0/0`
- Rule 3: Group 2 access to `0.0.0.0/0`
- Rule 4: Group 3 access to `0.0.0.0/0`
- Rule 5: Group 2 access to `172.131.0.0/16`

In this example, Rule 2, Rule 3, and Rule 4 are evaluated last. Group 1 has access to `10.1.0.0/16` only, and Group 2 has access to `172.131.0.0/16` only. Group 3 does not have access to `10.1.0.0/16` or `172.131.0.0/16`, but it has access to all other networks. If you remove Rules 1 and 5, all three groups have access to all networks.

Client VPN uses longest prefix matching when evaluating authorization rules. See Route priority in the *Amazon VPC User Guide* for more details.

Solution

Verify that you create authorization rules that explicitly grant Active Directory groups access to specific network CIDRs. If you add an authorization rule for `0.0.0.0/0`, keep in mind that it will be evaluated last, and that previous authorization rules may limit the networks to which it grants access.

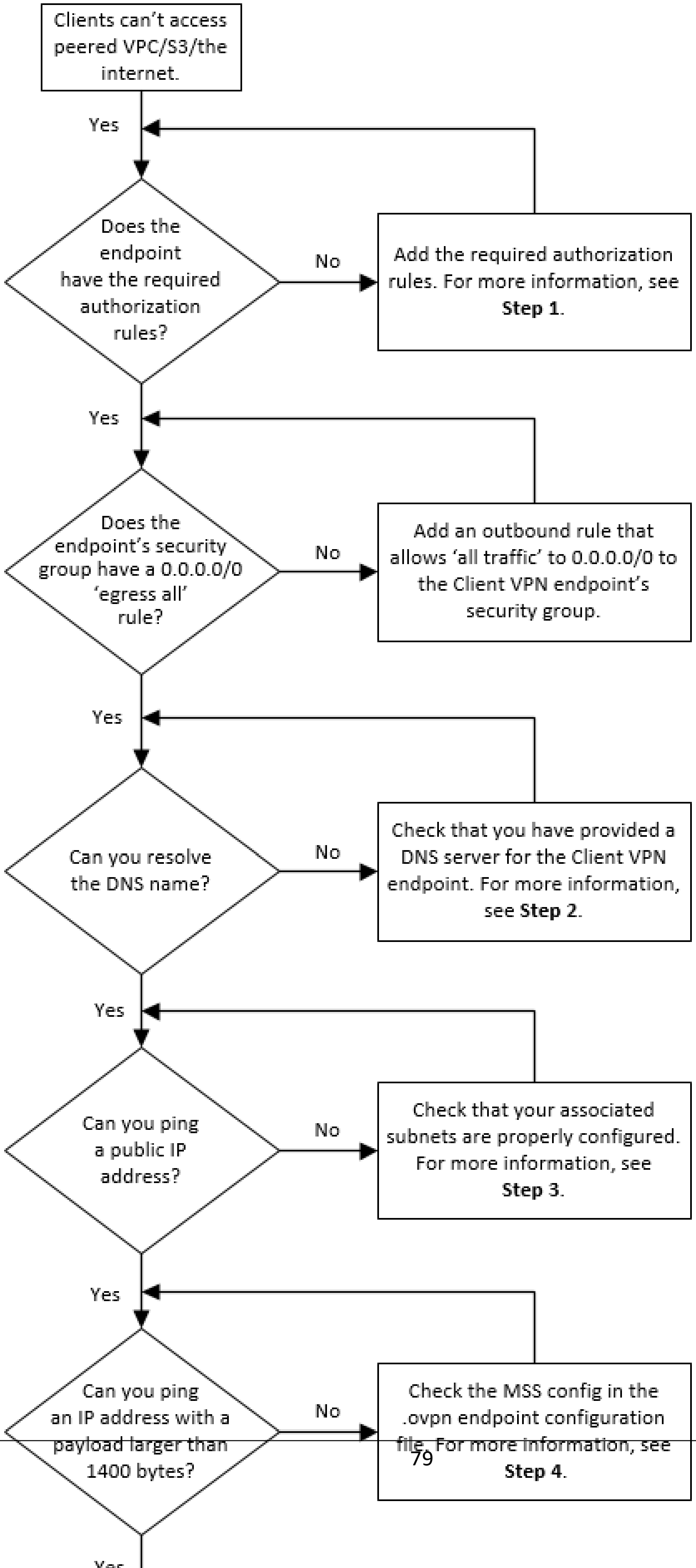
Clients can't access a peered VPC, Amazon S3, or the internet

Problem

I have properly configured my Client VPN endpoint routes, but my clients can't access a peered VPC, Amazon S3, or the internet.

Solution

The following flow chart contains the steps to diagnose internet, peered VPC, and Amazon S3 connectivity issues.


Clients can’t access peered VPC/S3/the internet.
Yes
Does the endpoint have the required authorization rules?
No
Add the required authorization rules. For more information, see **Step 1**.
Yes
Does the endpoint’s security group have a 0.0.0.0/0 ‘egress all’ rule?
No
Add an outbound rule that allows ‘all traffic’ to 0.0.0.0/0 to the Client VPN endpoint’s security group.
Yes
Can you resolve the DNS name?
No
Check that you have provided a DNS server for the Client VPN endpoint. For more information, see **Step 2**.
Yes
Can you ping a public IP address?
No
Check that your associated subnets are properly configured. For more information, see **Step 3**.
Yes
Can you ping an IP address with a payload larger than 1400 bytes?
No
Check the MSS config in the .ovpn endpoint configuration file. For more information, see **Step 4**.
Yes

1. For access to the internet, add an authorization rule for `0.0.0.0/0`.

   For access to a peered VPC, add an authorization rule for the IPv4 CIDR range of the VPC.

   For access to S3, specify the IP address of the Amazon S3 endpoint.
2. Check whether you are able to resolve the DNS name.

   If you are unable to resolve the DNS name, verify that you have specified the DNS servers for the Client VPN endpoint. If you manage your own DNS server, specify its IP address. Verify that the DNS server is accessible from the VPC.

   If you're unsure about which IP address to specify for the DNS servers, specify the VPC DNS resolver at the .2 IP address in your VPC.
3. For internet access, check if you are able to ping a public IP address or a public website, for example, `amazon.com`. If you do not get a response, make sure that the route table for the associated subnets has a default route that targets either an internet gateway or a NAT gateway. If the route is in place, verify that the associated subnet does not have network access control list rules that block inbound and outbound traffic.

   If you are unable to reach a peered VPC, verify that the associated subnet's route table has a route entry for the peered VPC.

   If you are unable to reach Amazon S3, verify that the associated subnet's route table has a route entry for the gateway VPC endpoint.
4. Check whether you can ping a public IP address with a payload larger than 1400 bytes. Use one of the following commands:

   - Windows

     ```
     C:\> ping  8.8.8.8 -l 1480 -f
     ```
   - Linux

     ```
     $ ping -s 1480 8.8.8.8 -M do
     ```

   If you cannot ping an IP address with a payload larger than 1400 bytes, open the Client VPN endpoint `.ovpn` configuration file using your preferred text editor, and add the following.

   ```
   mssfix 1328
   ```

Access to a peered VPC, Amazon S3, or the internet is intermittent

Problem

I have intermittent connectivity issues when connecting to a peered VPC, Amazon S3, or the internet, but access to associated subnets is unaffected. I need to disconnect and reconnect in order to resolve the connectivity issues.

Cause

Clients connect to a Client VPN endpoint based on the DNS round-robin algorithm. This means that their traffic can be routed through any of the associated subnets when they establish a connection. Therefore, they might experience connectivity issues if they land on an associated subnet that does not have the required route entries.

Solution

Verify that the Client VPN endpoint has the same route entries with targets for each associated network. This ensures that clients have access to all routes regardless of the associated subnet through which their traffic is routed.

For example, say that your Client VPN endpoint has three associated subnets (Subnet A, B, and C), and you want to enable internet access for your clients. To do this, you must add three `0.0.0.0/0` routes - one that targets each associated subnet:

- Route 1: `0.0.0.0/0` for Subnet A
- Route 2: `0.0.0.0/0` for Subnet B
- Route 3: `0.0.0.0/0` for Subnet C

Client software returns TLS error

Problem

I used to be able to connect my clients to the Client VPN successfully, but now the OpenVPN-based client returns the following error when it tries to connect:

```
TLS Error: TLS key negotiation failed to occur within 60 seconds (check your network
 connectivity)
TLS Error: TLS handshake failed
```

Possible causes

If you use mutual authentication and you imported a client certificate revocation list, the client certificate revocation list might have expired. During the authentication phase, the Client VPN endpoint checks the client certificate against the client certificate revocation list that you imported. If the client certificate revocation list has expired, you cannot connect to the Client VPN endpoint.

Alternatively, there might be an issue with the OpenVPN-based software that the client is using to connect to the Client VPN.

Solution

Check the expiry date of your client certificate revocation list by using the OpenSSL tool.

```
$ openssl crl -in path_to_crl_pem_file -noout -nextupdate
```

The output displays the expiry date and time. If the client certificate revocation list has expired, you must create a new one and import it to the Client VPN endpoint. For more information, see Client certificate revocation lists (p. 43).

For more information about troubleshooting OpenVPN-based software, see Troubleshooting Your Client VPN Connection in the *AWS Client VPN User Guide*.

Client software returns user name and password errors (Active Directory authentication)

Problem

I use Active Directory authentication for my Client VPN endpoint and I used to be able to connect my clients to the Client VPN successfully. But now, clients are getting invalid user name and password errors.

Possible causes

If you use Active Directory authentication and if you enabled multi-factor authentication (MFA) after you distributed the client configuration file, the file does not contain the necessary information to prompt users to enter their MFA code. Users are prompted to enter their user name and password only, and authentication fails.

Solution

Download a new client configuration file and distribute it to your clients. Verify that the new file contains the following line.

```
static-challenge "Enter MFA code " 1
```

For more information, see Export and configure the client configuration file (p. 55). Test the MFA configuration for your Active Directory without using the Client VPN endpoint to verify that MFA is working as expected.

Clients cannot connect (mutual authentication)

Problem

I use mutual authentication for my Client VPN endpoint. Clients are getting TLS key negotiation failed errors and timeout errors.

Possible causes

The configuration file that was provided to the clients does not contain the client certificate and the client private key, or the certificate and key are incorrect.

Solution

Ensure that the configuration file contains the correct client certificate and key. If necessary, fix the configuration file and redistribute it to your clients. For more information, see Export and configure the client configuration file (p. 55).

Client returns a credentials exceed max size error (federated authentication)

Problem

I use federated authentication for my Client VPN endpoint. When clients enter their user name and password in the SAML-based identity provider (IdP) browser window, they get an error that the credentials exceed the maximum supported size.

Cause

The SAML response returned by the IdP exceeds the maximum supported size. For more information, see Requirements and considerations for SAML-based federated authentication (p. 12).

Solution

Try to reduce the number of groups that the user belongs to in the IdP, and try connecting again.

Client does not open browser (federated authentication)

Problem

I use federated authentication for my Client VPN endpoint. When clients try to connect to the endpoint, the client software does not open a browser window, and instead displays a user name and password popup window.

Cause

The configuration file that was provided to the clients does not contain the `auth-federate` flag.

Solution

Export the latest configuration file (p. 55), import it to the AWS provided client, and try connecting again.

Client returns no available ports error (federated authentication)

Problem

I use federated authentication for my Client VPN endpoint. When clients try to connect to the endpoint, the client software returns the following error:

```
The authentication flow could not be initiated. There are no available ports.
```

Cause

The AWS provided client requires the use of TCP port 35001 to complete authentication. For more information, see Requirements and considerations for SAML-based federated authentication (p. 12).

Solution

Verify that the client's device is not blocking TCP port 35001 or is using it for a different process.

Verify the bandwidth limit for a Client VPN endpoint

Problem

I need to verify the bandwidth limit for a Client VPN endpoint.

Cause

The throughput depends on multiple factors, such as the capacity of your connection from your location, and the network latency between your Client VPN desktop application on your computer and the VPC endpoint.

Solution

Run the following commands to verify the bandwidth.

```
sudo iperf3 -s -V
```

On the client:

```
sudo iperf -c server IP address -p port -w 512k -P 60
```

Document history

The following table describes the AWS Client VPN Administrator Guide updates.

update-history-change	update-history-description	update-history-date
VPN session maximum duration	You can configure a shorter maximum VPN session duration to meet security and compliance requirements.	January 20, 2022
Client login banner	You can enable a text banner on AWS provided Client VPN desktop applications when a VPN session is established to meet regulatory and compliance needs.	January 20, 2022
Client connect handler	You can enable the client connect handler for your Client VPN endpoint to run custom logic that authorizes new connections.	November 4, 2020
Self-service portal	You can enable a self-service portal on your Client VPN endpoint for your clients.	October 29, 2020
Client-to-client access	You can enable clients that connect to a Client VPN endpoint to connect to each other.	September 29, 2020
SAML 2.0-based federated authentication	You can authenticate Client VPN users using SAML 2.0-based federated authentication.	May 19, 2020
Specify security groups during creation	You can specify a VPC and security groups when you create your AWS Client VPN endpoint.	March 5, 2020
Configurable VPN ports	You can specify a supported VPN port number for your AWS Client VPN endpoint.	January 16, 2020
Support for multi-factor authentication (MFA)	Your AWS Client VPN endpoint supports MFA if it's enabled for your Active Directory.	September 30, 2019
Support for split-tunnel	You can enable split-tunnel on your AWS Client VPN endpoint.	July 24, 2019
Initial release (p. 85)	This release introduces AWS Client VPN.	December 18, 2018