

Managed Dialup VPN with custom SSO Authentication

11 June 2021 - 6 min. read

[AWS Client VPN](#)

[AWS Single Sign-On](#)

[Dialup VPN](#)

[Single-Sign-On \(SSO\)](#)

In these days of remote and smart working giving users access to private resources and applications is a hot topic.

A dial-up VPN is the tool that can solve this problem, giving home users and road warriors access to private corporate services that are not exposed on the internet (even if they are hosted on the AWS Cloud).

Implementing a VPN for end users is always a non-trivial task, because there are always opposite requirements like:

- Ease of configuration on clients and servers
- Security
- Centralized management

AWS offers the **AWS Client VPN** service that can help you to give remote access to resources in a VPC and leverage external identity providers to authenticate users such as Okta, Active Directory and other services using the SAML protocol.

AWS Client VPN users can connect to a self-service web portal, download client software and the configuration needed to connect to the private resource, easing the effort needed to implement the solution because there's no need for an administrator to be involved in the process.

Example scenario

Some time ago we released an article on [how to implement single sign on on the aws console using G Suite as an identity provider](#)

Based on the considerations made in the previous article about different IdPs we want to configure the AWS Client VPN service using G Suite as the authentication provider, unfortunately there's a catch that we're still investigating.

The issue is that the AWS Client VPN software uses a plain http service to authenticate requests, while G Suite accepts and validates only https addresses (we'll see some details about configuration later).

We'll set up AWS SSO as an authentication provider, so we'll be able in the future to switch the user database quite easily and finally configure G Suite as our Identity Source.

AWS SSO is also useful if you're using AWS Organizations to manage a multi-account scenario to give users different access to specific accounts, you can find [some topology examples here](#).

We are going to use a default SSO setup using the internal authentication, the default setup is available at [this link](#).

In our example configuration we'll give users VPN access to a VPC in an development account:

VPC Name: test-vpc

VPC CIDR: 172.31.0.0/16

Client network CIDR: 172.20.20.0/22 (must not overlap with the destination CIDR or any other network that need to be reached using the VPN connection)

We'll go through different steps:

- Define SAML applications for self-service portal and vpn authentication
- Define Identity providers for self-service ad vpn client
- Create a Client VPN Endpoint
- Associate subnets, configure Authorization and allow traffic through Security Groups

- Test the configuration

Define SAML applications

We need to define SAML applications that our Client VPN will trust to authenticate users, one for the Self-Service Portal and the other for Client VPN application.

On the Organization root account Console go to “**AWS Single Sign-On**”, choose “**Applications**”, “**Add a new application**” and “**Add a custom SAML 2.0 Application**”

Add New Application

Choose an application from our catalog of preintegrated cloud applications or choose to add a custom SAML 2.0 application. Each application comes with detailed instructions to help you set up the trust between AWS SSO and the application's service provider. [Learn more](#)

AWS SSO Application Catalog

Type the name of an application

- Add a custom SAML 2.0 application
You can add SSO integration to your custom SAML 2.0-enabled applications

We'll name it “SSO Client VPN Self Service Portal”

Configure SSO Client VPN Self Service Portal

AWS SSO works as an identity provider (IdP) for any SAML 2.0-compliant cloud applications. To configure this application for SSO access, you must establish a trust relationship between AWS SSO and your cloud application (service provider) through a SAML metadata exchange. You can view instructions on this page and find metadata details for your provider.

[View instructions](#)

Details

Display name* SSO Client VPN Self Service Portal ⓘ

Description Application for client vpn self service portal

The description you type here does not appear in the user portal. However, it will be visible in the AWS SSO console and when using the AWS SSO APIs.

AWS SSO metadata

Your cloud application may require the following certificate and metadata details to recognize AWS SSO as the identity provider.

AWS SSO SAML metadata file	https://portal.sso.eu-west-1.amazonaws.com/saml	Copy URL	Download
AWS SSO sign-in URL	https://portal.sso.eu-west-1.amazonaws.com/saml	Copy URL	
AWS SSO sign-out URL	https://portal.sso.eu-west-1.amazonaws.com/saml	Copy URL	
AWS SSO issuer URL	https://portal.sso.eu-west-1.amazonaws.com/saml	Copy URL	
AWS SSO certificate	Download certificate		

Application properties

Your cloud application may optionally take additional settings to configure your user experience. [Learn more](#)

Application start URL ⓘ

Relay state

Session duration* 1 hour

Application metadata

AWS SSO requires specific metadata about your cloud application before it can trust this application. You can type this metadata manually or upload a metadata exchange file.

Application ACS URL* [/ice.clientvpn.amazonaws.com/api/auth/sso/saml](https://self-service.clientvpn.amazonaws.com/api/auth/sso/saml) ⓘ

Application SAML audience* urn:amazon:webservices:clientvpn

If you have a metadata file, you can upload it now instead.

* Required fields

[Cancel](#) [Save changes](#)

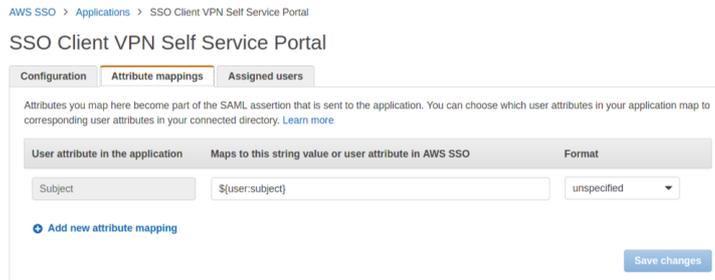
Click on the “**Download**” link for the AWS SSO SAML metadata file and **keep it secret**

Select “**Manually type your metadata values**” and fill in these values:

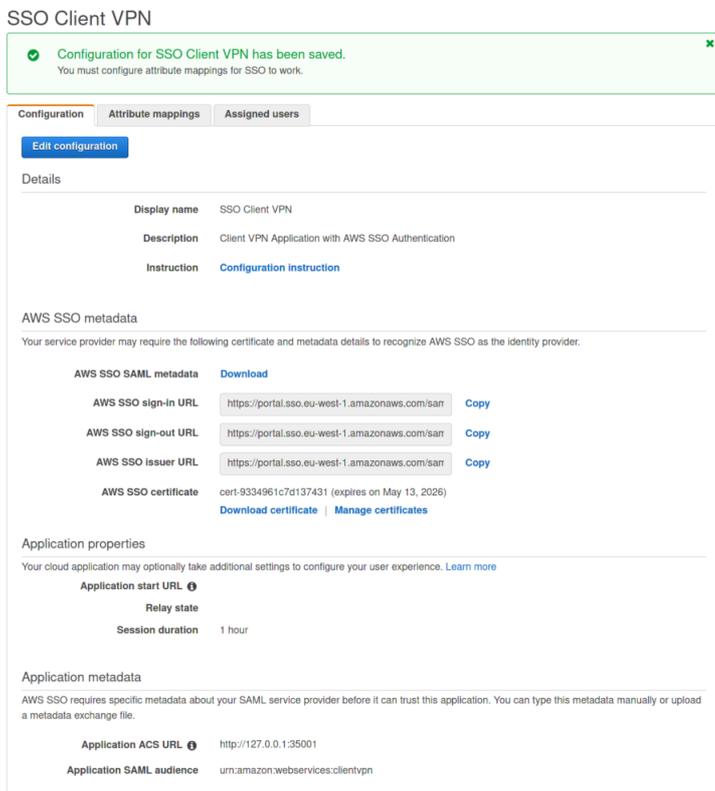
Application ACS URL: <https://self-service.clientvpn.amazonaws.com/api/auth/sso/saml>

Application SAML Audience: urn:amazon:webservices:clientvpn

After adding the application select “Attributes mappings” and map the `${user:subject}` attribute to the Subject field



After adding the Self service Portal application we'll need to add the VPN Client application:



Use this values:

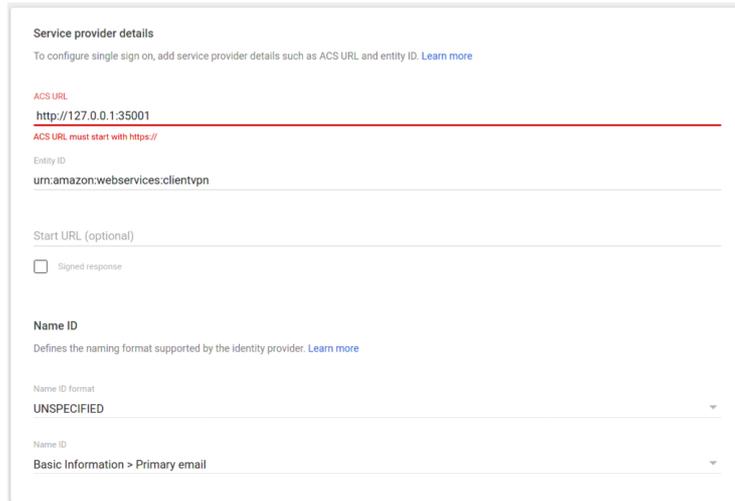
Application ACS URL: http://127.0.0.1:35001

Application SAML Audience: urn:amazon:webservices:clientvpn

Download this metadata and **keep it secret**

In this case you can see that the application ACS URL is something you wouldn't expect: <http://127.0.0.1>.

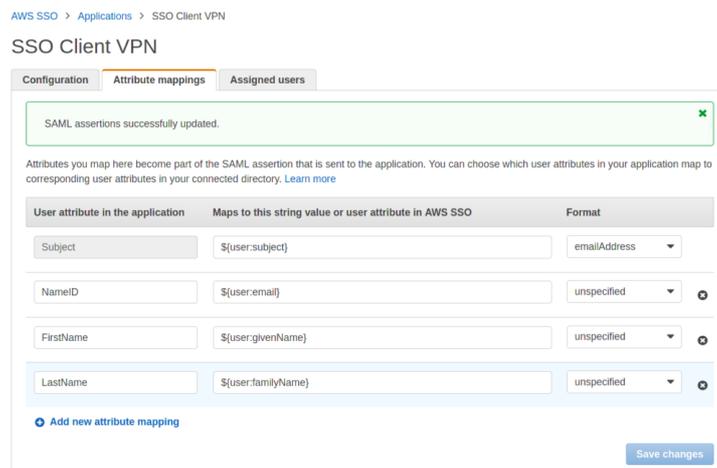
This is because the VPN Client application will spawn a service listening on the client to be able to validate and forward SAML assertions. This is the reason because G Suite authentication needs further investigation, if you configure the SAML application in G Suite you'll get this validation error:



The screenshot shows a 'Service provider details' form. At the top, it says 'To configure single sign on, add service provider details such as ACS URL and entity ID. [Learn more](#)'. Below this, there is a red error message: 'ACS URL must start with https://'. The ACS URL field contains 'http://127.0.0.1:35001'. The Entity ID field contains 'urn:amazon:webservices:clientvpn'. There is a 'Start URL (optional)' field and a checkbox for 'Signed response'. The 'Name ID' section has a 'Name ID format' dropdown set to 'UNSPECIFIED' and a 'Name ID' dropdown set to 'Basic Information > Primary email'.

Spoiler: we've been able to overcome this behavior with a little hack! We are going to write a separated article soon for this, so keep following us!

After adding the Client VPN application select Attributes mappings and map them:



The screenshot shows the 'Attribute mappings' tab for an 'SSO Client VPN' application. A green notification box at the top says 'SAML assertions successfully updated.' Below this, there is a table for mapping user attributes. The table has three columns: 'User attribute in the application', 'Maps to this string value or user attribute in AWS SSO', and 'Format'. The rows are: Subject (maps to \${user:subject}, format: emailAddress), NameID (maps to \${user:email}, format: unspecified), FirstName (maps to \${user:givenName}, format: unspecified), and LastName (maps to \${user:familyName}, format: unspecified). There is an 'Add new attribute mapping' link and a 'Save changes' button.

User attribute in the application	Maps to this string value or user attribute in AWS SSO	Format
Subject	<input style="width: 100%;" type="text" value="\${user:subject}"/>	emailAddress
NameID	<input style="width: 100%;" type="text" value="\${user:email}"/>	unspecified
FirstName	<input style="width: 100%;" type="text" value="\${user:givenName}"/>	unspecified
LastName	<input style="width: 100%;" type="text" value="\${user:familyName}"/>	unspecified

If you map them incorrectly the authentication will fail (please pay attention the the format of the Subject field, you need to change it to “**emailAddress**”)

After adding SAML applications we'll need to tell the destination account (development in our case) to trust them as an identity provider (don't forget to assign users or groups to your applications ! using the “Assigned users” tab, otherwise no application will be available after the user logs in)

Define Identity providers for self-service portal and VPN client

Log into the development account console, go to **IAM** -> **Identity Providers** and click on “**Add Provider**”

Select SAML, add a provider name (we'll use *clientvpn-sso-idp* for client vpn application and *clientvpn-portal-idp* for self-service portal application), choose the previously downloaded metadata files and upload them

The screenshot shows the AWS IAM console interface for creating a new identity provider. On the left is a navigation sidebar for 'Identity and Access Management (IAM)'. The main content area is titled 'IAM > Identity providers > Create Identity Provider' and 'Add an Identity provider'. Under 'Configure provider', there are two radio button options for 'Provider type': 'SAML' (selected) and 'OpenID Connect'. Below this is a text input for 'Provider name' with a character limit of 128. A 'Metadata document' section includes a 'Choose file' button and a note that the file must be a valid UTF-8 XML document. An 'Add tags (Optional)' section shows 'No tags associated with the resource.' and an 'Add tag' button. At the bottom right are 'Cancel' and 'Add provider' buttons.

We are now ready to create the VPN Client endpoint in our VPC and configure it to trust our SAML applications.

Create a Client VPN Endpoint

For the client vpn endpoint you'll need a wildcard ACM Certificate associated with your domain, if you don't have one create it before creating the Endpoint, refer to [this documentation](#) to create it.

In the Development account go to **VPC** and select "**Client VPN Endpoints**", create a new client vpn endpoint.

Create a new Client VPN endpoint to enable clients to access networks over a TLS VPN session

Name Tag: dev-client-vpn ⓘ

Description: client vpn for dev account ⓘ

Client IP4 CIDR: 172.20.20.0/22 ⓘ

Authentication Information

Server certificate ARN: arn:aws:acm:eu-west-1:046933179291:certificate ⓘ

Authentication Options: Choose one or more authentication methods from below ⓘ

Use mutual authentication

Use user-based authentication

Connection Logging

Do you want to log the details on client connections? Yes ⓘ No ⓘ

Client Connect Handler

Do you want to enable Client Connect Handler? Yes ⓘ No ⓘ

Other Optional Parameters

DNS Server 1 IP address: ⓘ

DNS Server 2 IP address: ⓘ

Transport Protocol: TCP ⓘ UDP ⓘ

Enable split-tunnel: ⓘ

VPC ID: vpc-7ccc6c05 ⓘ

Security Group IDs: sg-54b5cafa ⓘ

Select security groups ⓘ

Group ID	Group Name	VPC ID	Description
sg-54b5cafa	default	vpc-7ccc6c05	default VPC security group

VPN port: 443 ⓘ

Enable self-service portal: ⓘ

Select **“Use user-based authentication”** and select the previously created IdPs:

Authentication Information

Server certificate ARN: arn:aws:acm:eu-west-1:046933179291:certificate ⓘ

Authentication Options: Choose one or more authentication methods from below ⓘ

Use mutual authentication

Use user-based authentication

Active Directory authentication

Federated authentication

SAML provider ARN: arn:aws:iam:046933179291:saml-provider/clientv ⓘ

Self-service SAML provider ARN: .046933179291:saml-provider/clientvpn-portal-idp ⓘ

Don't forget to tick the **“Enable self-service portal”** checkbox

After saving the configuration please copy the self service portal url and modify the “SSO Client VPN service portal” application to use it as “Application start URL, otherwise the user will not be able to access the self-service portal:

```

Connection log true
Cloudwatch log group clientvpn-logs
Cloudwatch log stream log
Client IP4 CIDR 172.20.20.0/22
SAML provider ARN arn:aws:iam:046933179291:saml-provider/clientvpn-ssoidp
Self-service SAML provider ARN arn:aws:iam:046933179291:saml-provider/clientvpn-portal-idp
Client certificate ARN
Transport protocol udp
Split-tunnel Enabled
VPC ID vpc-7ccc6c05
Self-service portal URL https://self-service.clientvpn.amazonaws.com/endpoints/cvpn-endpoint-015bbd5ee638c1164
Client Connect Handler State applied

```

SSO Client VPN Self Service Portal

Configuration Attribute mappings Assigned users

Edit configuration

Details

Display name SSO Client VPN Self Service Portal

Description Application for client vpn self service portal

Instruction [Configuration instruction](#)

AWS SSO metadata

Your service provider may require the following certificate and metadata details to recognize AWS SSO as the identity provider.

AWS SSO SAML metadata [Download](#)

AWS SSO sign-in URL <https://portal.sso.eu-west-1.amazonaws.com/san> [Copy](#)

AWS SSO sign-out URL <https://portal.sso.eu-west-1.amazonaws.com/san> [Copy](#)

AWS SSO issuer URL <https://portal.sso.eu-west-1.amazonaws.com/san> [Copy](#)

AWS SSO certificate cert-9401b30658aee354 (expires on May 13, 2026) [Download certificate](#) | [Manage certificates](#)

Application properties

Your cloud application may optionally take additional settings to configure your user experience. [Learn more](#)

Application start URL <https://self-service.clientvpn.amazonaws.com/endpoints/cvpn-endpoint-015bbd5ee638c1164>

Relay state

Session duration 1 hour

Application metadata

AWS SSO requires specific metadata about your SAML service provider before it can trust this application. You can type this metadata manually or upload a metadata exchange file.

Application ACS URL <https://self-service.clientvpn.amazonaws.com/api/auth/sso/saml>

Application SAML audience urn:amazon:webservices:clientvpn

Associate subnets, configure Authorization and allow traffic through Security Groups

Click the “Associations” tab on the client vpn endpoint, select the target VPC and the subnet to associate them, after some time they will show as associated:

Client VPN Endpoint: cvpn-endpoint-015bbd5ee638c1164

Summary Associations Security Groups Authorization Route Table Connections Tags

Associate Disassociate

Filter by attributes

Association ID	Network ID	Description	Endpoint ID	State	Security Groups
cvpn-assoc-02...	subnet-2d899b...	-	cvpn-endpoint-...	Associated	2 Security Groups
cvpn-assoc-01...	subnet-166a35...	-	cvpn-endpoint-...	Associated	2 Security Groups
cvpn-assoc-02...	subnet-5827ca...	-	cvpn-endpoint-...	Associated	2 Security Groups

Click on the Authorization tab and authorize the VPC network segment:

Client VPN Endpoint: cvpn-endpoint-015bbd5ee638c1164

Summary Associations Security Groups Authorization Route Table Connections Tags

Authorize Ingress Revoke Ingress

Filter by attributes

Endpoint ID	Description	Group ID	Access all	Destination Cidr	State
cvpn-endpoint-015bbd5ee638c1164	vpc	-	true	172.31.0.0/16	Active

You'll also see that route tables will automatically be populated:

Client VPN Endpoint: cvpn-endpoint-015bbd5ee638c1164

Summary Associations Security Groups Authorization Route Table Connections Tags

Create Route Delete Route

Filter by attributes or search by keyword

Endpoint ID	Destination Cidr	Target Subnet	Type	Origin	State	Description
cvpn-endpoint-...	172.31.0.0/16	subnet-2d899b...	Nat	associate	Active	Default Route
cvpn-endpoint-...	172.31.0.0/16	subnet-166a35...	Nat	associate	Active	Default Route
cvpn-endpoint-...	172.31.0.0/16	subnet-5827ca...	Nat	associate	Active	Default Route

Test the configuration

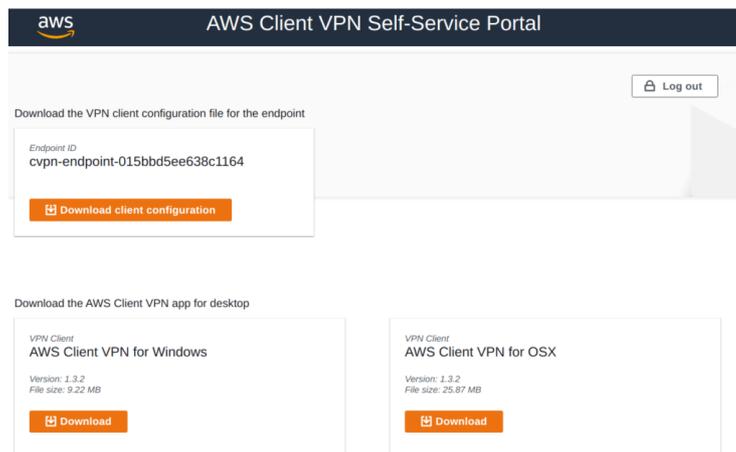
Open a browser and go to the user portal url for your SSO application, you can find it in the “Settings” page of SSO configuration, something like: <https://example->

org.awsapps.com/start

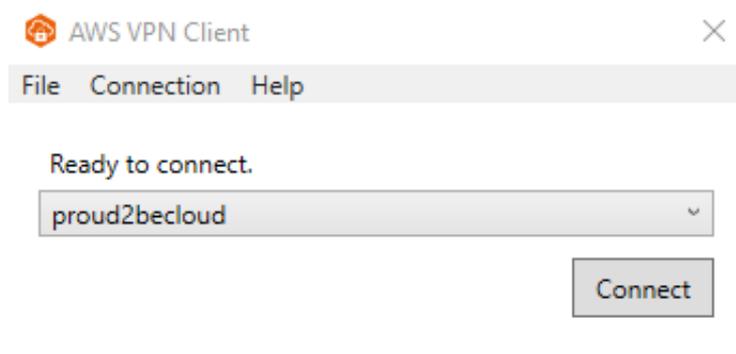
After logging you'll see the list of configured applications:



Select the SSO Client VPN Self Service to download the configuration file and client software.



After installing software import the configuration (file -> manage profiles -> add profile)



Click on "connect", a new browser window will open asking for credentials. After logging in a confirmation window will appear:



The client will be in the "connected state" and you'll see an entry in the "Connections" tab

Client VPN Endpoint: cvpn-endpoint-015bbd5ee638c1164

Summary Associations Security Groups Authorization Route Table **Connections** Tags

Terminate Connection

Filter by attributes

Connection ID	Client VPN Endp	Timeest	Usernam	Connector	Posturi	Ingress E	Egress I	Ingre	Egre	Client IP	Com	Status	Conne
cvpn-connect...	cvpn-endpoint...	2021...	N/A	2021-05...	-	0	0	0	0	-	-	Terminated	2021-05...
cvpn-connect...	cvpn-endpoint...	2021...	demia...	2021-05...	-	11296	2558	17	14	172.20.2...	-	Active	-

On the client you'll see that routing tables are added automatically to reach your VPC:

```

ca Command Prompt

C:\Users\test>route print

Interface List
=====
5...00 ff 6c 16 0a d9 .....AWS VPN Client TAP-Windows Adapter V9
7...08 00 27 dd 40 bd .....Intel(R) PRO/1000 MT Desktop Adapter
1.....Software Loopback Interface 1
=====

IPv4 Route Table
=====
Active Routes:
Network Destination        Netmask          Gateway          Interface        Metric
-----
0.0.0.0                    0.0.0.0          10.0.2.2         10.0.2.15        25
10.0.2.0                   255.255.255.0   On-link         10.0.2.15        281
10.0.2.15                  255.255.255.255 On-link         10.0.2.15        281
10.0.2.255                 255.255.255.255 On-link         10.0.2.15        281
127.0.0.0                  255.0.0.0       On-link         127.0.0.1        331
127.0.0.1                  255.255.255.255 On-link         127.0.0.1        331
127.255.255.255           255.255.255.255 On-link         127.0.0.1        331
172.20.21.160             255.255.255.224 On-link         172.20.21.162    257
172.20.21.162             255.255.255.255 On-link         172.20.21.162    257
172.20.21.191             255.255.255.255 On-link         172.20.21.162    257
172.31.0.0                 255.255.0.0     172.20.21.161   172.20.21.162    1
224.0.0.0                  240.0.0.0       On-link         127.0.0.1        331
224.0.0.0                  240.0.0.0       On-link         10.0.2.15        281
224.0.0.0                  240.0.0.0       On-link         172.20.21.162    257
255.255.255.255           255.255.255.255 On-link         127.0.0.1        331
255.255.255.255           255.255.255.255 On-link         10.0.2.15        281
255.255.255.255           255.255.255.255 On-link         172.20.21.162    257
=====
Persistent Routes:
None

IPv6 Route Table
=====
Active Routes:
If Metric Network Destination      Gateway
-----
1 331 ::1/128 On-link
7 281 fe80::/64 On-link
5 281 fe80::/64 On-link
7 281 fe80::3180:45f:3d81:c4e8/128 On-link
5 281 fe80::6534:8a5f:6a3a:c2f8/128 On-link
1 331 ff00::/8 On-link
7 281 ff00::/8 On-link
5 281 ff00::/8 On-link
=====
Persistent Routes:
None

```

Wrap-Up

AWS Client VPN is a managed service that eases the task of configuring vpn connections for end-users, it offers a lot of out-of-the-box configuration mechanisms; in this article, we explored a custom implementation that is not described in the official documentation.

We're still searching for the best way to add G Suite authentication, we're investigating having G Suite as an identity source for AWS SSO and then using SSO SAML applications to map the right attributes for the IDP and Client VPN. Add your thoughts about it in the comments! See you again in 14 days on **#Proud2beCloud!**



Damiano Giorgi

Ex on-prem systems engineer, lazy and prone to automating boring tasks. In constant search of technological innovations and new exciting things to experience. And that's why I love Cloud Computing! At this moment, the only "hardware" I regularly dedicate myself to is that my bass; if you can't find me in the office or in the band room try at the pub or at some airport, then!



Simone Merlini

CEO and co-founder of beSharp, Cloud Ninja and early adopter of any type of *aaS solution. I divide myself between the PC keyboard and the one with black and white keys; I specialize in deploying gargantuan dinners and testing vintage bottles.
