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)

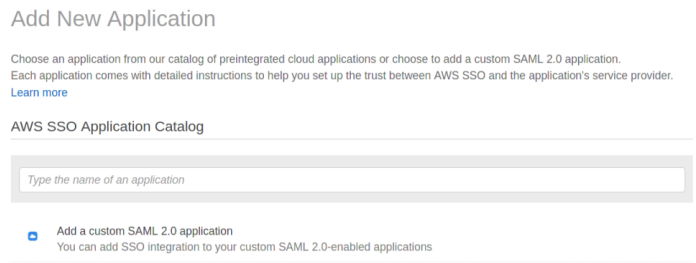
We'll go through different steps:

- 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

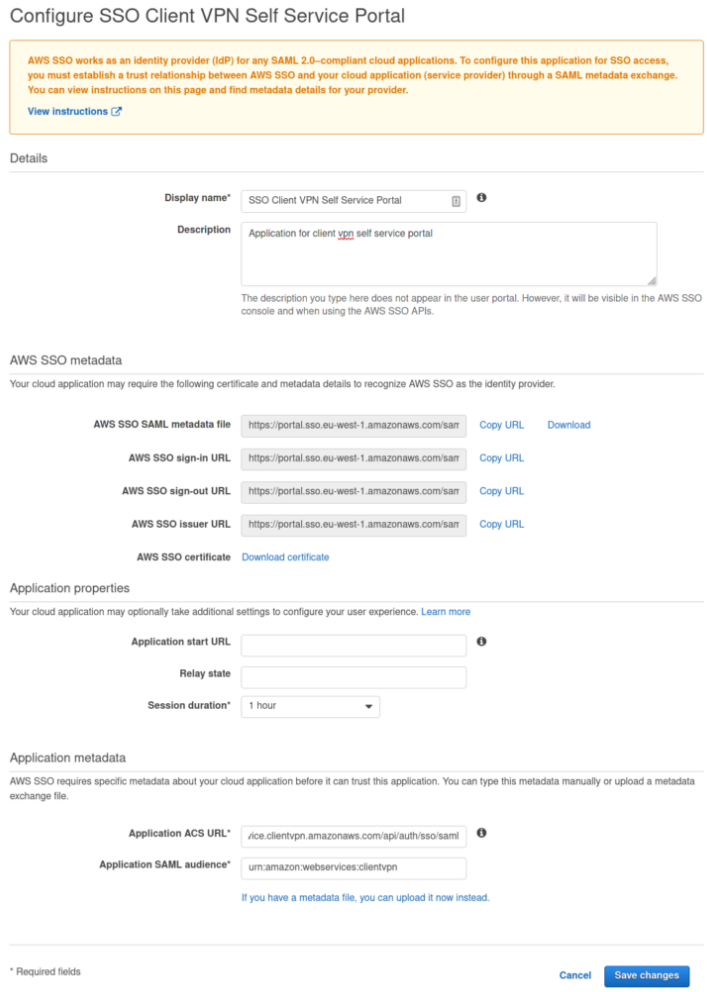
- Test the configuration

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**"



We'll name it "SSO Client VPN Self Service Portal"



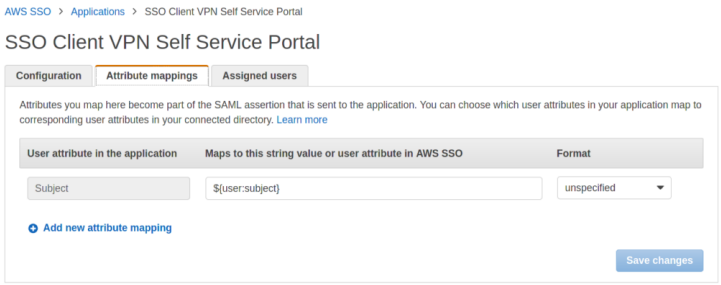
Click on the "**Download**" link for the AWS SSO SAML metadata file and **keep it secret**

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

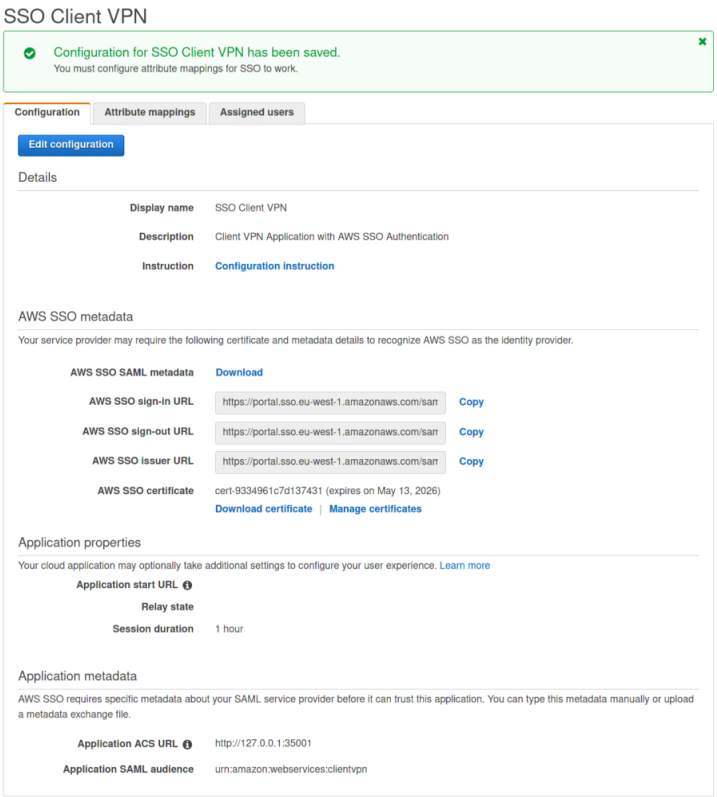
Application ACS URL: https://self-service.clientvpn.amazonaws.com/api/auth/sso/saml

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:

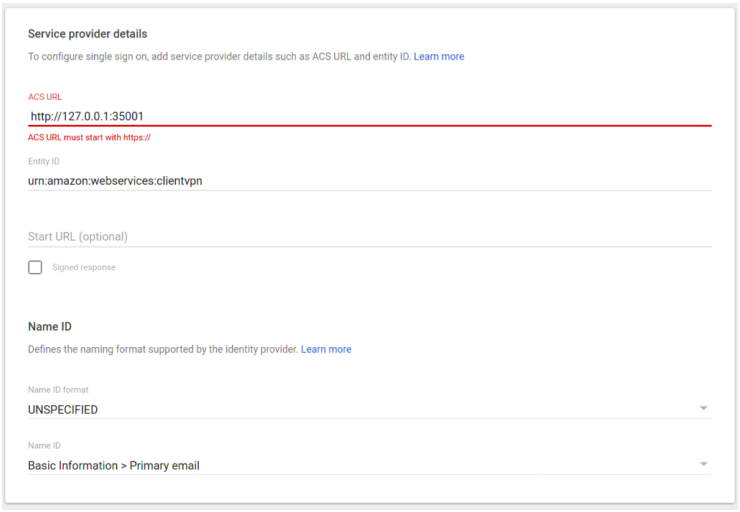
Application ACS URL: http://127.0.0.1:35001

Application SAML Audience: urn:amazon:webservices:clientvpn

Download this metadata and **keep it secret**

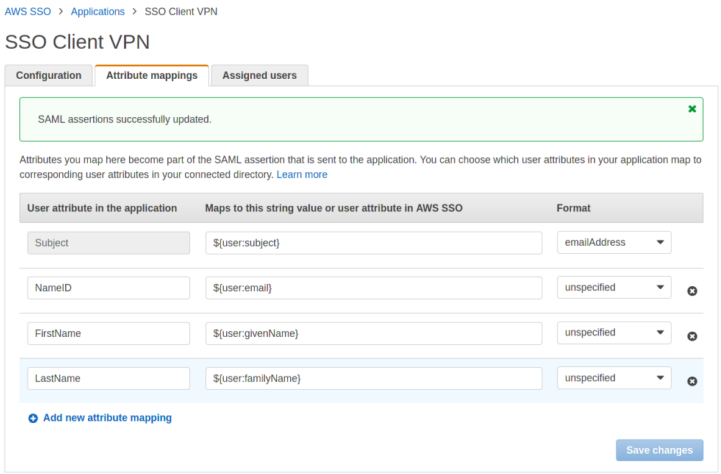
In this case you can see that the application ACS URL is something you wouldn’t expect: http://127.0.0.1.

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:



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:



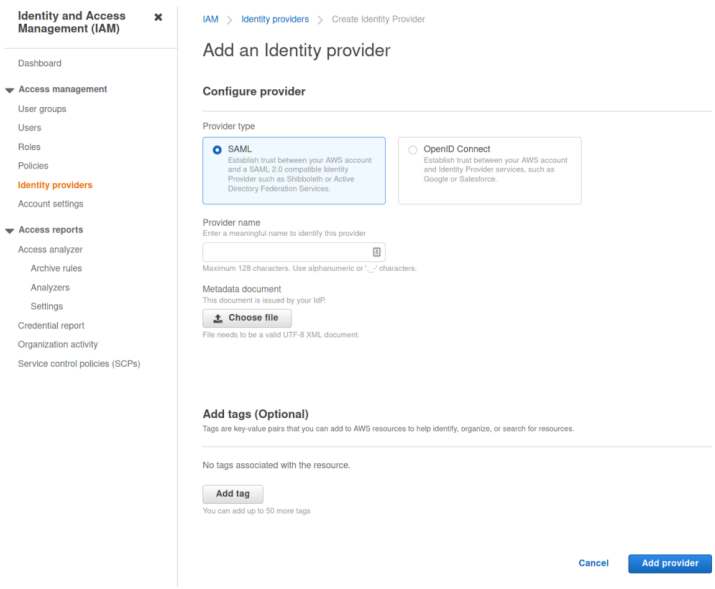
If you map them incorrectly the authentication will fail (please pay attention the the format of the Subject field, you need to change it to "**emailAddress**")

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 ad 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

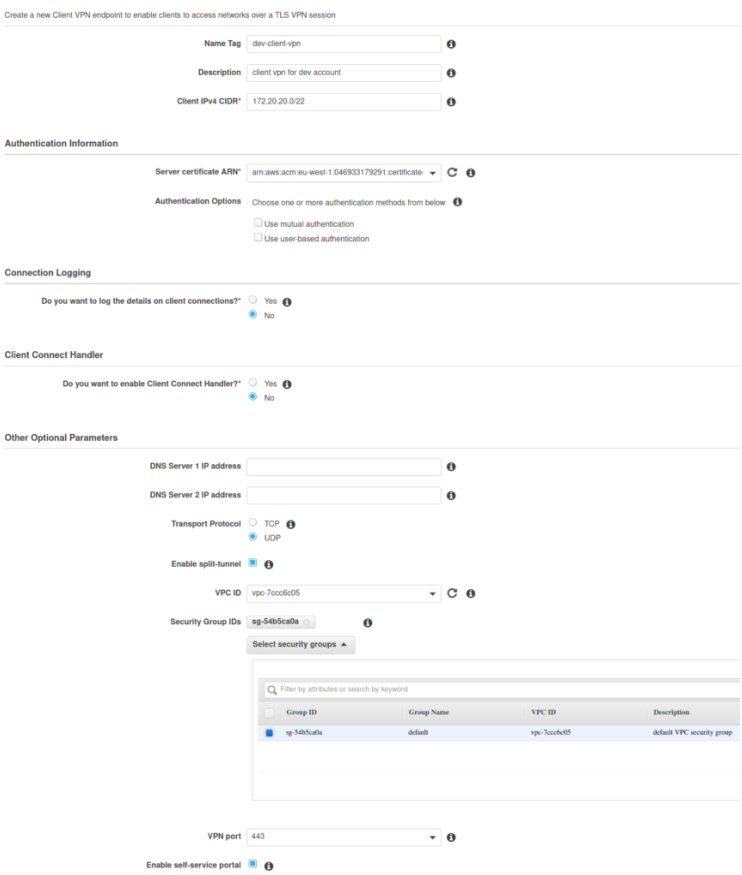


We are now ready to create the VPN Client endpoint in our VPC and configure it to trust our SAML applications.

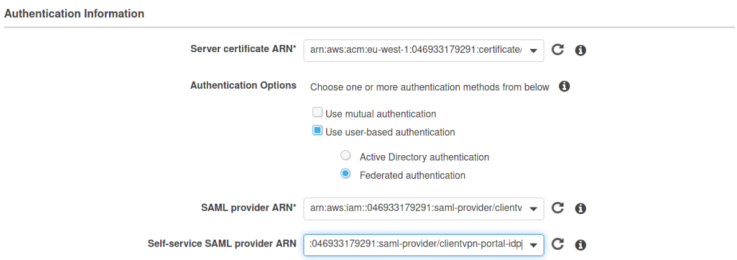
Create a Client VPN Endpoint

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.



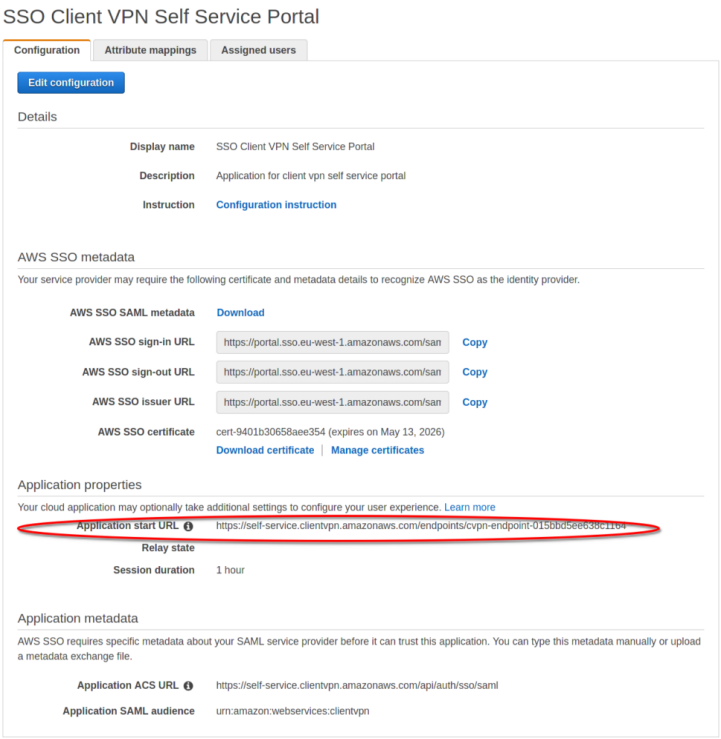
Select “**Use user-based authentication**” and select the previously created IdPs:



Don’t forget to thick the “**Enable self-service portal”** checkbox

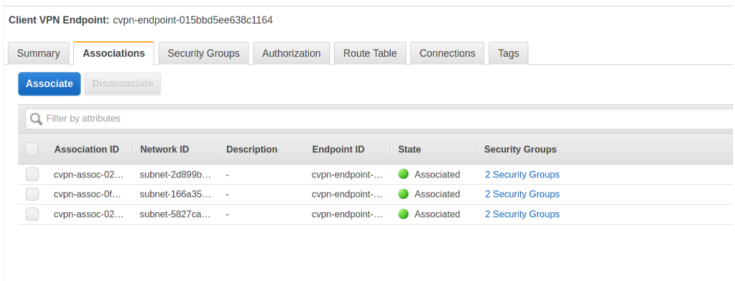
After saving the configuration please copy the self service portal url and modify the “SSO Client VPN service portal” application to use it as “Application start URL, otherwise the user will not be able to access the self-service portal:

Connection log	true
Cloudwatch log group	clienvpn-logs
Cloudwatch log stream	log
Client IPv4 CIDR	172.20.20.0/22
SAML provider ARN	arn:aws:iam::046933179291:saml-provider/clientvpn-sso-idp
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

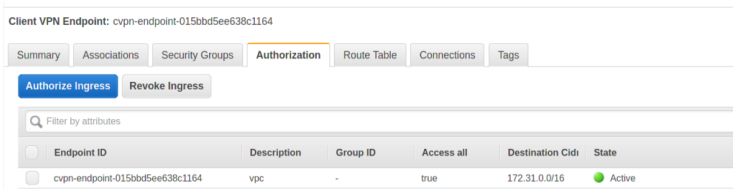


Associate subnets, configure Authorization and allow traffic through Security Groups

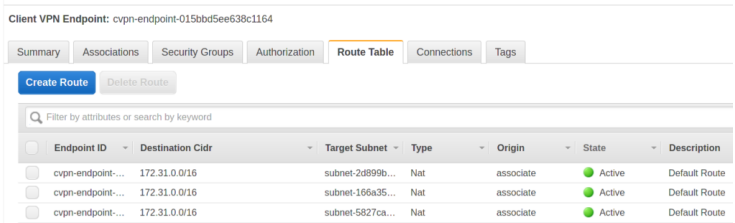
Click the “Associations” tab on the client vpn endpoint, select the target VPC and the subnet to associate them, after some time they will show as associated:



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



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

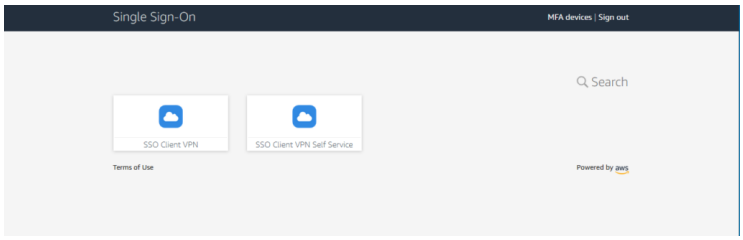


Test the configuration

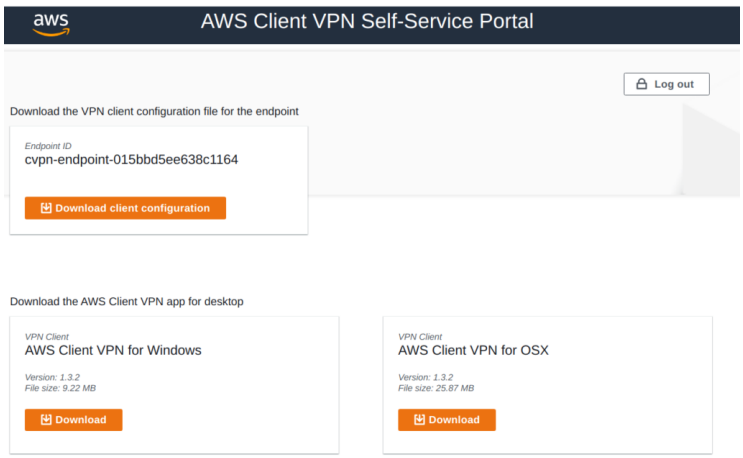
Open a browser and go to the user portal url for your SSO application, you can find it in the “Settings” page of SSO configuration, something like: https://example-

org.awsapps.com/start

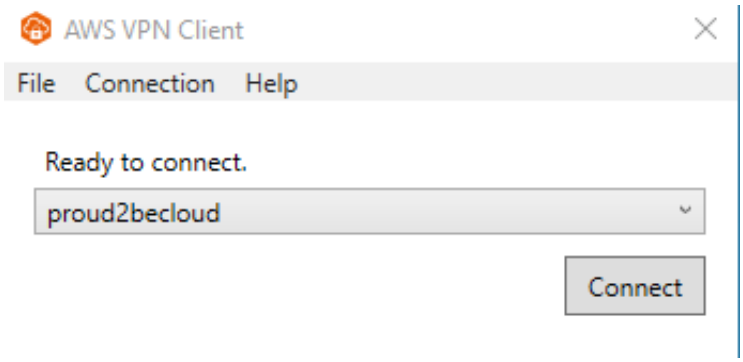
After logging you'll see the list of configured applications:



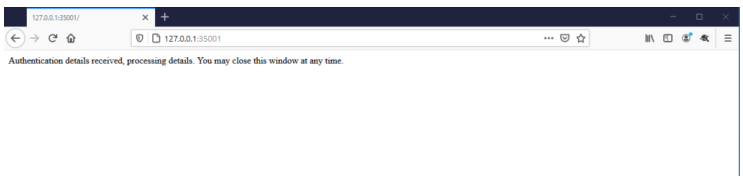
Select the SSO Client VPN Self Service to download the configuration file and client software.



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

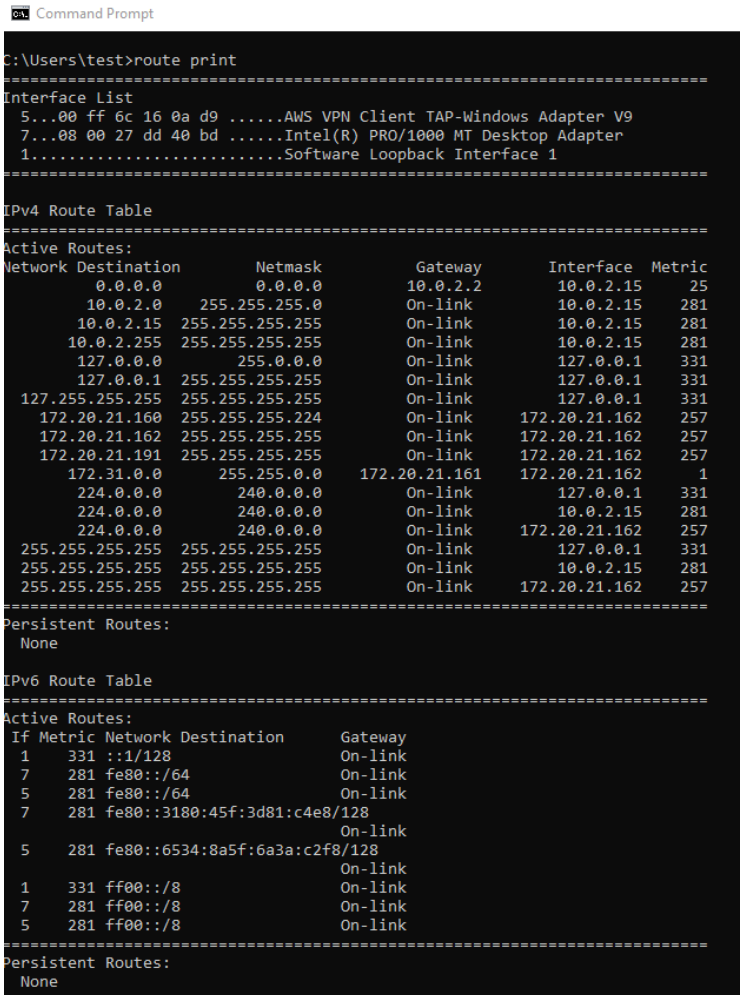


Click on "connect", a new browser window will open asking for credentials. After logging in a confirmation window will appear:



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

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



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.
