AWS Client VPN SAML authentication with Google G-Suite

Posted on December 15, 2020
by Vallard (https://benincosa.com/?author=1)

Note: Video for this Blog Post is Here (https://youtu.be/kYJaHankFAg).

When dealing with cloud resources the two opposing needs are security and accessibility. When we often deploy resources in a private network inside of an AWS VPC that are not accessible directly from the outside. To access these resources, we can use a bastion server or VPN.

The bastion server is a server that is accessible on the public network but also has a connection into the private network. The drawbacks of of a bastion server is that you would have to login to it to do your work. Most of my development is on my laptop and I like using the tools on my laptop as opposed to some server that may not have the tools. We can configure SSH tunnels, but this can also be a hassle.

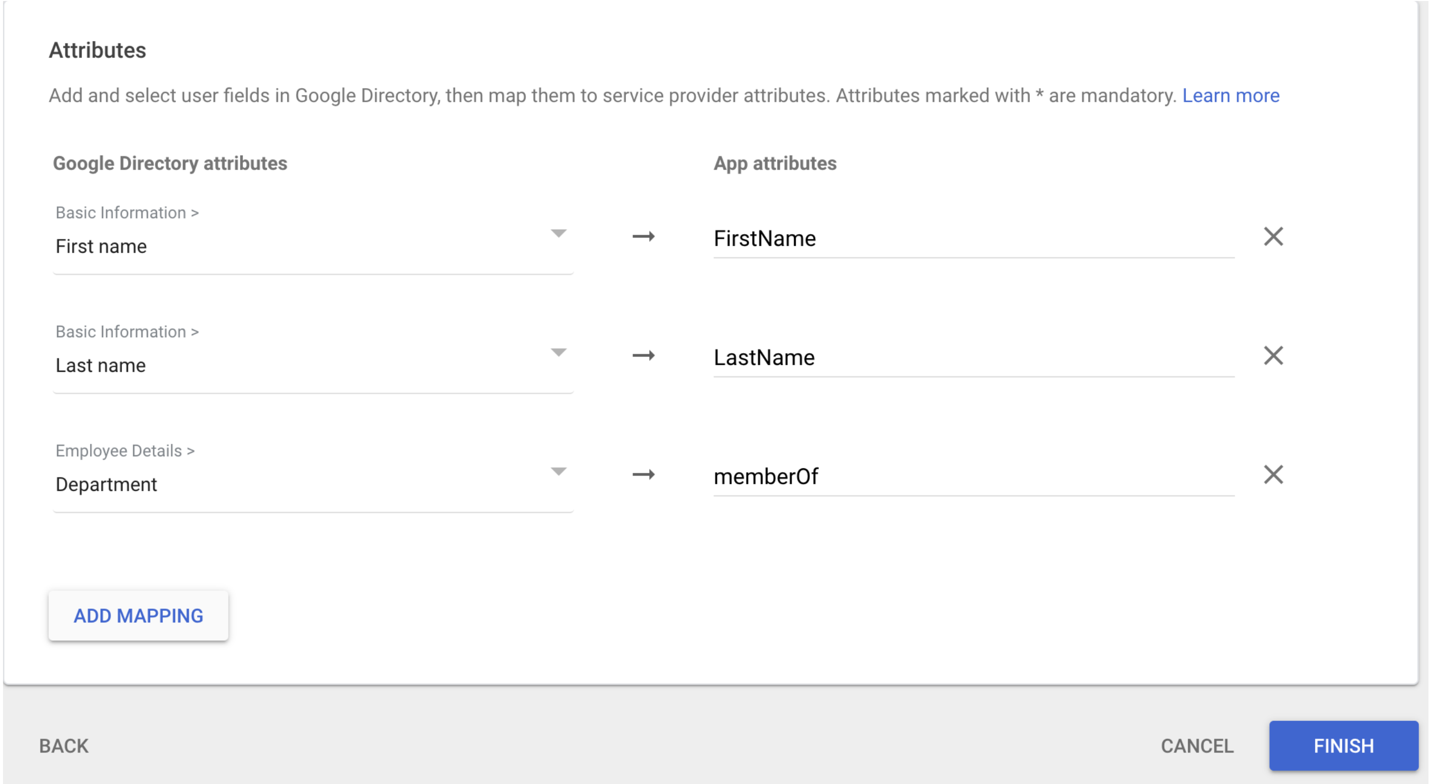
AWS has a managed Client VPN service that allows us to access this remote network and internal resources pretty easily. I'm usually not a fan of client VPNs as it reminds me of big corporation life. But the fact that I can tweak it and manage it with split tunnel and authenticate with my company GMail account makes it a pretty cool option.

1. Google G Suite Setup

Special shout out to Stack Overflow (https://stackoverflow.com/questions/63151685/aws-vpn-using-federated-login-with-google-idp-app-not-configured-for-user/63590967#63590967) for helping with this as it was the hardest part!

1. In your G Suite Admin account go to the Admin Console (https://admin.google.com/?hl=en). Select Apps.
2. Select SAML apps
3. Add custom SAML app
4. Give it a name like Castle Rock AWS
5. Under Service Provider details:
    1. ACS URL: `https://127.0.0.1:35001`
    2. Entity ID: `urn:amazon:webservices:clientvpn`
    3. Check Signed Response
    4. Name ID
        1. Name ID format: `UNSPECIFIED`
        2. Name ID: `Basic Information > Primary email`

You now need to add the mapping. The SAML docs tell you (https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/client-authentication.html) how this should be:



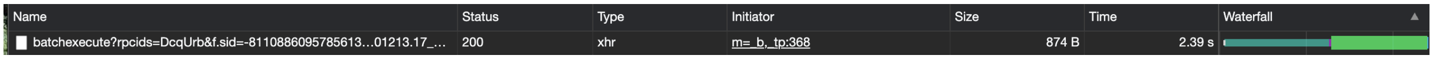
Attributes

Add and select user fields in Google Directory, then map them to service provider attributes. Attributes marked with * are mandatory. Learn more

Google Directory attributes		App attributes
Basic Information > First name	→	FirstName
Basic Information > Last name	→	LastName
Employee Details > Department	→	memberOf

1.1 Google Apps Correction

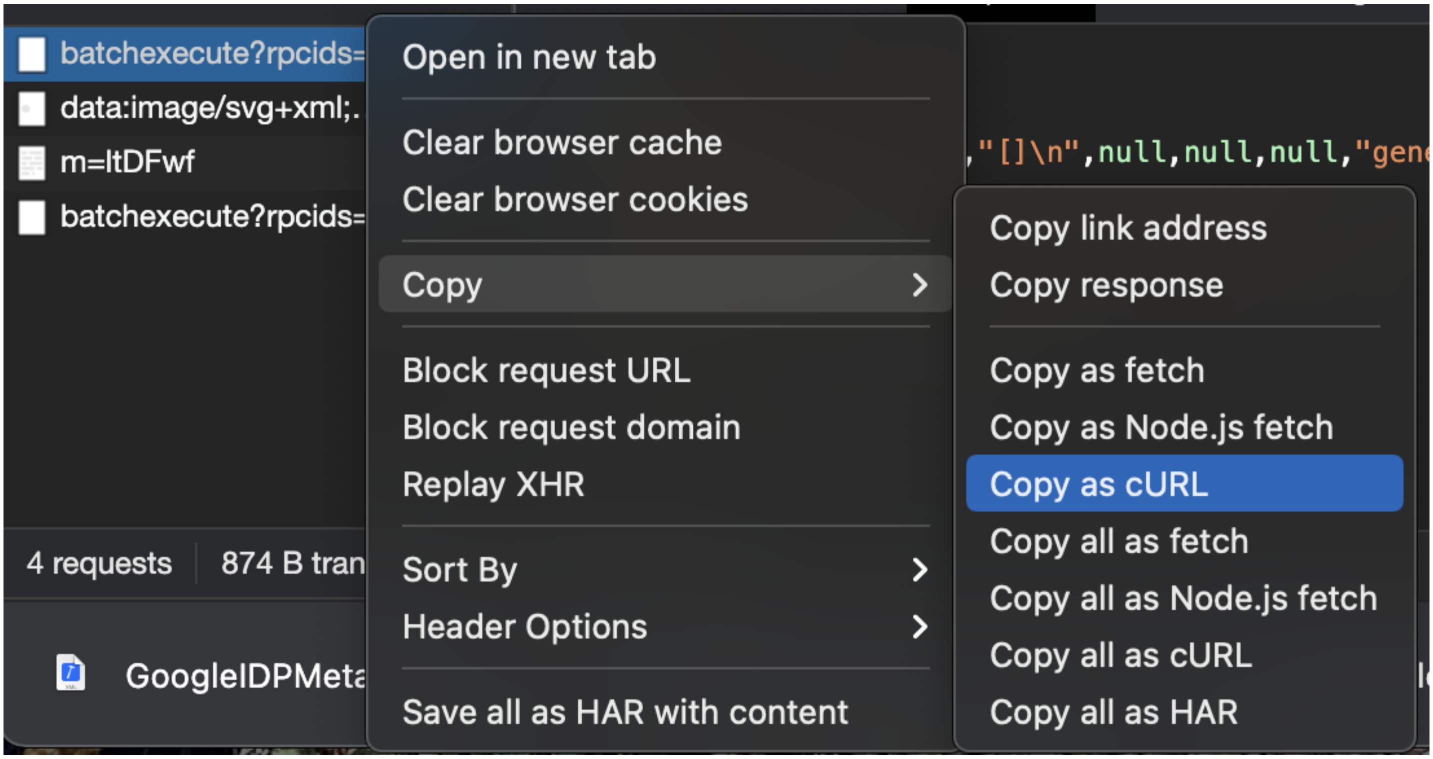
Google SAML requires that you use https but Amazon isn't having any of that. Amazon specifies it wants `http://127.0.0.1:35001` . In order to get around it, you have to hack Google. Once your new SAML app is created, go to edit the Service Provider Details. Change the 127.0.0.1 to something like 127.0.1.1. I do this in Google Chrome. Then open the console and navigate to the `Network` tab. Monitor the network and hit save. You should see some webpage load called `batchexecute` …



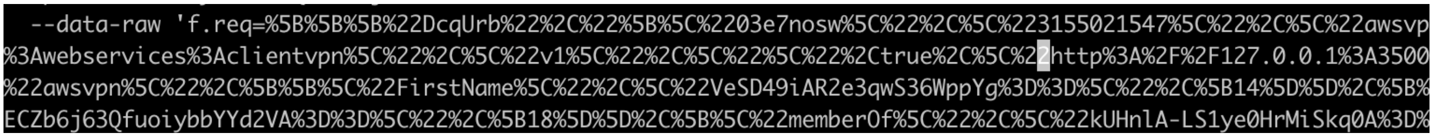
Name	Status	Type	Initiator	Size	Time	Waterfall
batchexecute?rpcids=DcqUrb&f.sid=-8110886095785613…01213.17_…	200	xhr	m=_b,_tp:368	874 B	2.39 s	

the first part that loads

Now, right click and copy as cURL



Open up a terminal and paste all that madness in it. (I saved my to `/tmp/foo`) Then find the 127 and change it to http as well as the 127.0.1.1 to 127.0.0.1



```
  --data-raw 'f.req=%5B%5B%5B%22DcqUrb%22%2C%22%5B%5C%2203e7nosw%5C%22%2C%5C%223155021547%5C%22%2C%5C%22awsvp
%3Awebservices%3Aclientvpn%5C%22%2C%5C%22v1%5C%22%2C%5C%22%5C%22%2Ctrue%2C%5C%22http%3A%2F%2F127.0.0.1%3A3500
%22awsvpn%5C%22%2C%5B%5B%5C%22FirstName%5C%22%2C%5C%22VeSD49iAR2e3qwS36WppYg%3D%3D%5C%22%2C%5B14%5D%5D%2C%5B%
ECZb6j63QfuoiybbYYd2VA%3D%3D%5C%22%2C%5B18%5D%5D%2C%5B%5C%22memberOf%5C%22%2C%5C%22kUHnlA-LS1ye0HrMiSkq0A%3D%
```

Next, run that curl command:

```
bash /tmp/foo
```

That should then set you up. Refreshing your G-Suite SAML apps dashboard, you should now see that it is set to `http://127.0.0.1`

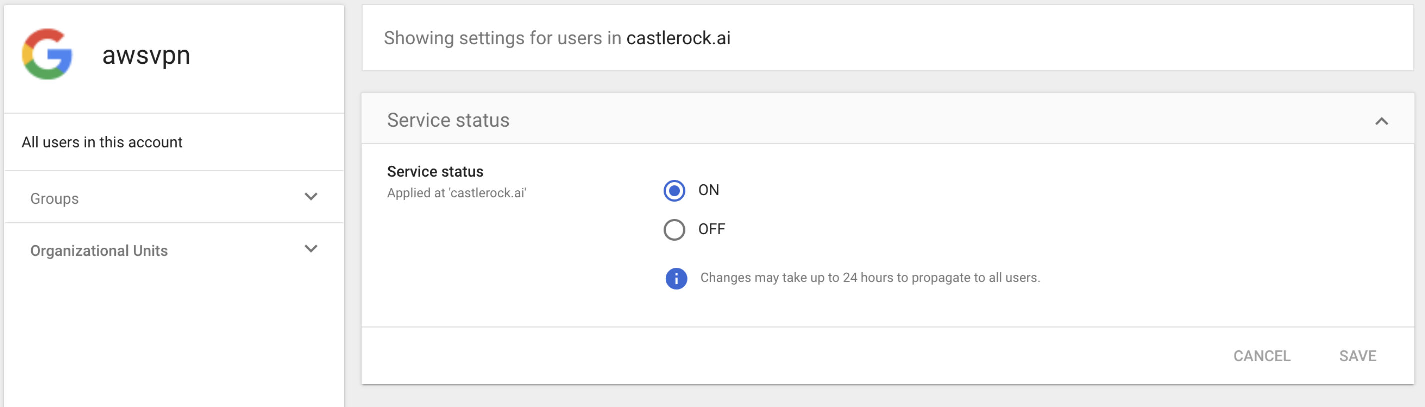
1.2 Create SAML in AWS

Download the Metadata XML file from Google. Then go to your downloads folder and run:

```
aws iam create-saml-provider --saml-metadata-document fileb://GoogleIDPMetadata.xml --na
me CastleRockSAML
```

1.3 Configure Users

You may have some users or all users you want to add into the VPN. You can configure this now at the Service Status. Make sure whoever you test with has access. The most simple is to give all users with your domain access. This works great for small companies.


awsvpn
All users in this account
Groups
Organizational Units
Showing settings for users in castlerock.ai
Service status
CANCEL
SAVE

Service status
Applied at 'castlerock.ai'

● ON
○ OFF

Changes may take up to 24 hours to propagate to all users.

2. Client Certificate Setup

The next step is you need to generate a server certificate for the VPN connection. We can follow some of the steps Amazon provides (https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/client-authentication.html) to do this.

```
git clone https://github.com/OpenVPN/easy-rsa.git
cd easy-rsa/easyrsa3
./easyrsa init-pki
./easyrsa build-ca nopass
./easyrsa build-server-full server nopass
```

We need the `pki/issued/server.crt` , `pki/private/server.key` , and the `pki/ca.crt` . With those we run:

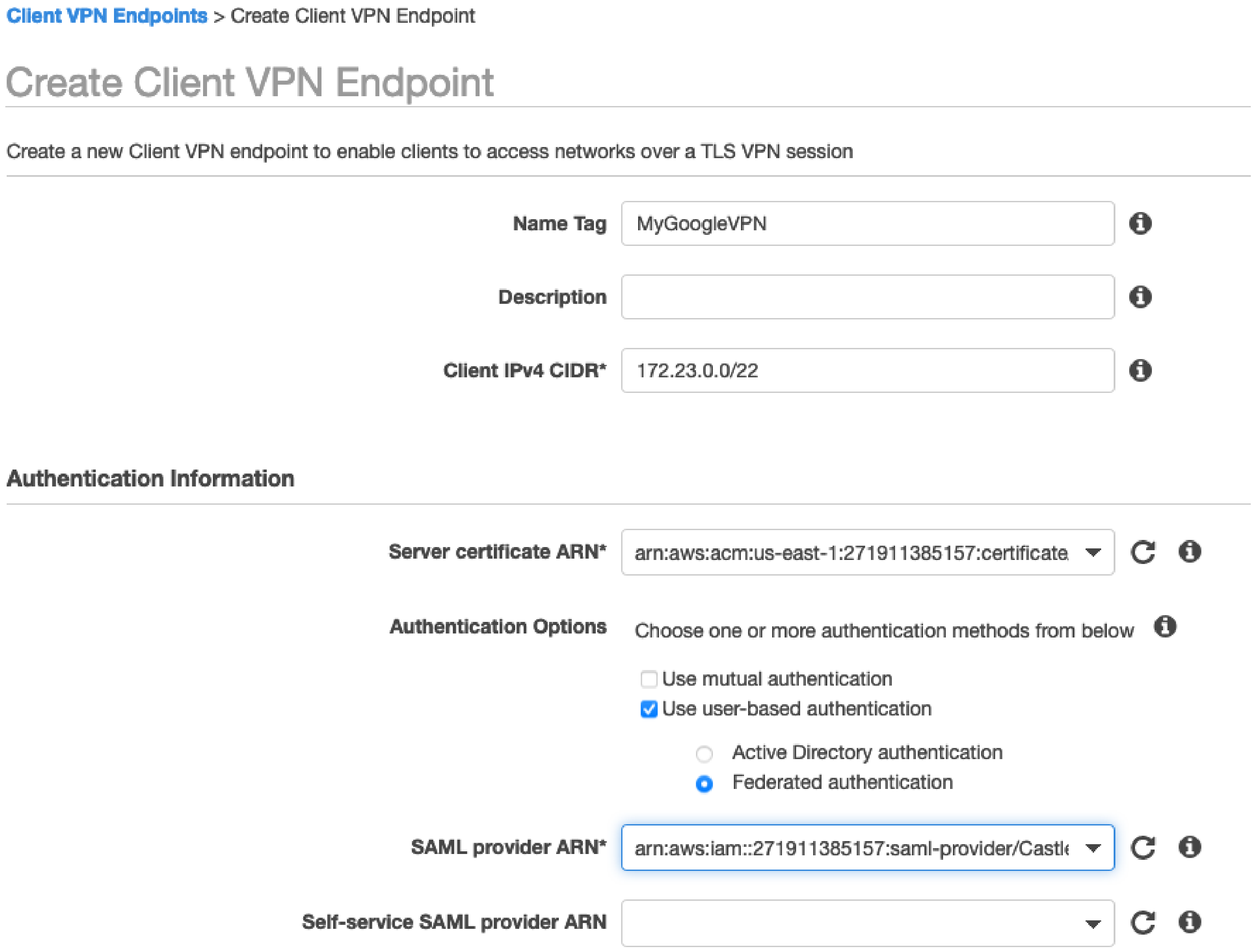
```
aws acm import-certificate --certificate fileb://server.crt --private-key fileb://server.key --certificate-chain fileb://ca.crt --region region
```

If you want to be able to authenticate without Google Mail for other users like contractors outside your company, you can generate the build-client-full instructions like Amazon says as well (https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/client-authentication.html).

At this point you should have the keys up in AWS Cert Manager. You are almost there!

3. Set up Client VPN Endpoint

We fill the form out with the following values



Client VPN Endpoints > Create Client VPN Endpoint

Create Client VPN Endpoint

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

Name Tag	MyGoogleVPN
Description	
Client IPv4 CIDR*	172.23.0.0/22

Authentication Information

Server certificate ARN*	arn:aws:acm:us-east-1:271911385157: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::271911385157:saml-provider/Castl
Self-service SAML provider ARN	

Our IPv4 CIDR should not overlap with any of the CIDRs in your VPC. My VPC is all 10.21.0.0/16 so 172.23.0.0/22 works pretty well. The netmask has to be between 16-22. We are using the self signed certificate we generated and the SAML provider we generated with the Google ID. The rest of the details you can leave as default or configure as you want. Be sure to enable split tunnel or your users will hate you.

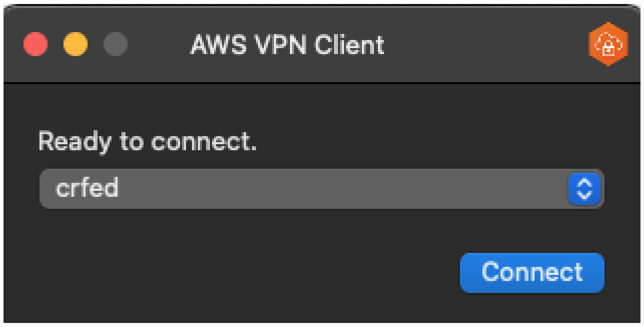
4. Configure VPN Connections

At this point all the hard work is done. You now just need to associate the VPN with the VPC you want it to join and add the subnets you want it to be able to reach. Set up routing and authorization as specified in the documentation (https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/cvpn-working.html). For my simple purposes I added the subnet to the VM I was looking to attach to. If I had an EKS cluster I'd have put it in the same subnets to be able to access the control plane.

5. Connect with VPN Client

You can download the VPN configuration from the Client VPN dashboard.

The AWS OpenVPN client can be downloaded from here (https://aws.amazon.com/vpn/client-vpn-download/). Importing the configuration our users will be presented with their Google SSO page to access the VPN.

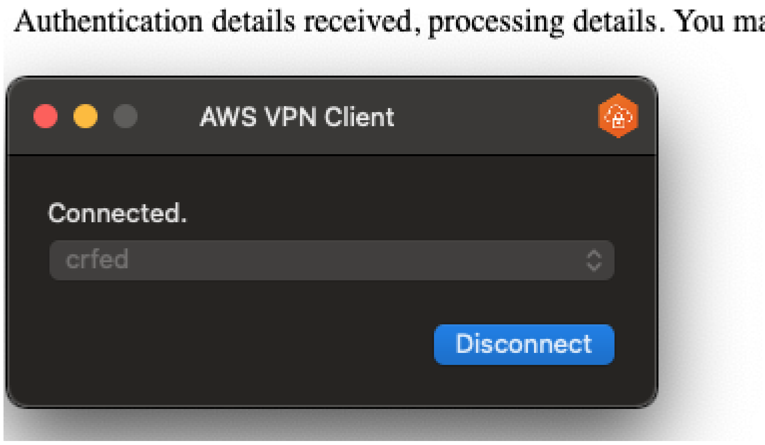


Ready to connect.

crfed

VPN Client

At this point, if we have configured the VPN to be able to access the subnet our VMs or resources we're interested in are on, we are able to connect to them without a bastion server. Connecting to Databases are simple as well and we do it in a secure way without exposing these services over the internet.


Authentication details received, processing details. You may close this window at any time.

Connected.

crfed

Our SSH config file has the following entry:

```
Host vpntest
  User ec2-user
  Hostname 10.21.101.93
  IdentityFile ~/.ssh/vallard01.pem
```

We simply connect with `ssh vpntest` and are immediately on the server. So easy! …Well at least it was after we set it all up.

📂 Application Architecture (https://benincosa.com/?cat=744)
AWS Client VPN (https://benincosa.com/?tag=aws-client-vpn), Gmail (https://benincosa.com/?tag=gmail), Google Mail (https://benincosa.com/?tag=google-mail), Gsuite (https://benincosa.com/?tag=gsuite), SAML (https://benincosa.com/?tag=saml), VPN (https://benincosa.com/?tag=vpn)

Vallard (https://benincosa.com/?author=1)

FullStack Developer. Former HPC traveling systems administrator. I write code, manage infrastructure, and design datacenters.

ALSO ON **VALLARD'S TECH BLOG**

New pastures – Vallard's Blog

9 years ago • 1 comment

Today is my last day as an employee at IBM. Its been fantastic. I've had a great …

Backing up UCS – Vallard's Blog

9 years ago • 3 comments

Backing up UCS can be a little confusing especially since it presents you a few …

Ethereum + Docker + Terraform + …

6 years ago • 1 comment

I created a quick way to get a private ethereum cluster up. I'll be presenting some of …

Subvie
iOS 8 w

7 years ago

That title
Basically,
know how

What do you think?

0 Responses

👍 Upvote 😝 Funny 😍 Love 😯 Surprised 😤 Angry 😥 Sad

Comments and reactions for this thread are now closed. ×

0 Comments Vallard's Tech Blog 🔒 Disqus' Privacy Policy 1 Login

♡ Favorite Tweet Share Sort by Best

This discussion has been closed.

✉ Subscribe Ⓓ Add Disqus to your siteAdd DisqusAdd ⚠ Do Not Sell My Data

Proudly powered by WordPress (https://wordpress.org/)