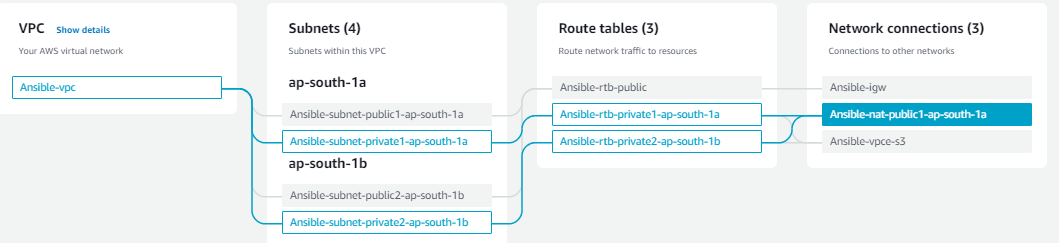
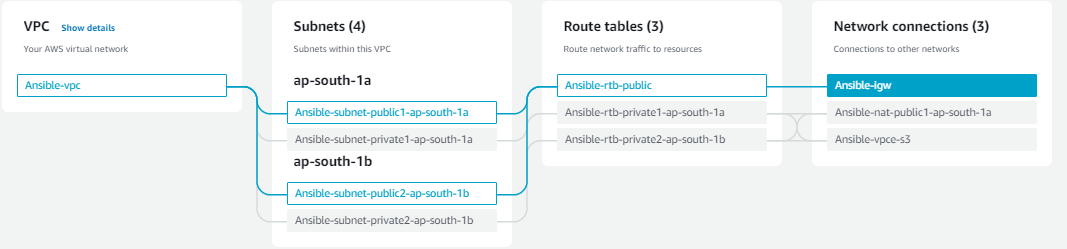
*Creat Ec2 instance with mfa authentication by using Ansible*

PART-1

# VPC CREATION

* Create VPC With CDIR Range 10.0.0.0/16
* Public Subnets (2) CIDR Range 10.0.0.0/24
* Privet Subnets (2) CIDR Range 10.0.1.0/24
* Create Route Table For VPC (2)
* Create inter Net Gateway (Public Subnets) And Nat Gateway For Private Subnets
* Assign Privet Route Table to Privet Subnets
* Assign Public Route Table to Public Subnets

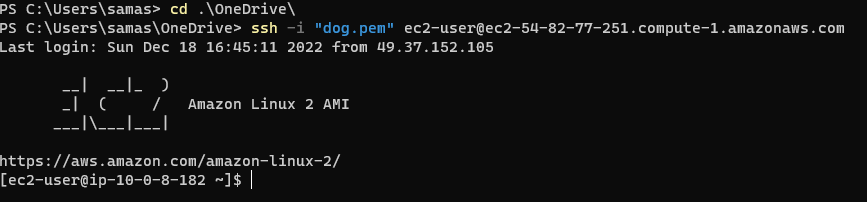




# EC2 INSTANCE CREATION

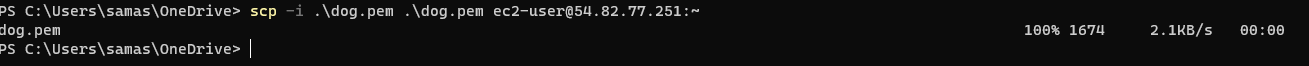
* Create Ec2 Instance (Public) Give The Name For Instance Ansible-Master
* Ami:Amazone Linux
* Instance Type : t2.micro
* Key Pair : .Pem
* Subnets : Public Subnets (VPC)
* Security Group : SSH(22),HTTP(80)
* Create Ec2 Instance (Privet) Give The Name For Instance Clinte
* Ami:Amazone Linux
* Instance Type : t2.micro
* Key Pair : .Pem
* Subnets : Privet Subnets (VPC)
* Security Group : SSH(22),
* Connect Ec2 Instance(Pub) With Terminal (Local) By Using This Command

***# ssh -i ".pem" ec2-user*** ***Public IPv4 DNS***



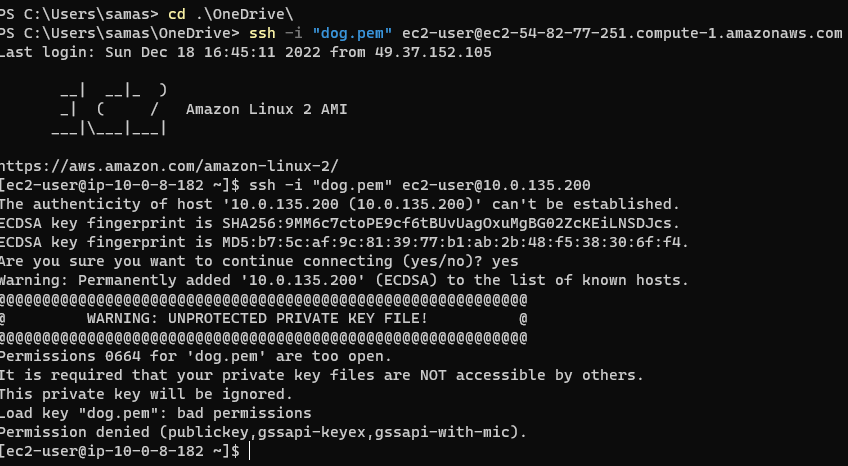
* In Terminal Open New Tab Copy dog.pem From Local Drive Paste To Ec2 Instance (Public) +IP
* By Using Command

# ***scp -i .\dog.pem .\dog.pem ec2-user@54.82.77.251:~***



* Connect Ec2 Instance(PVT) With Terminal (Local) By Using This Command

***# ssh -i ".pem" ec2-user*** ***PVT+IP***



How to setup Two Factor Authentication for SSH in AWS EC2 – AWS MFA Setup

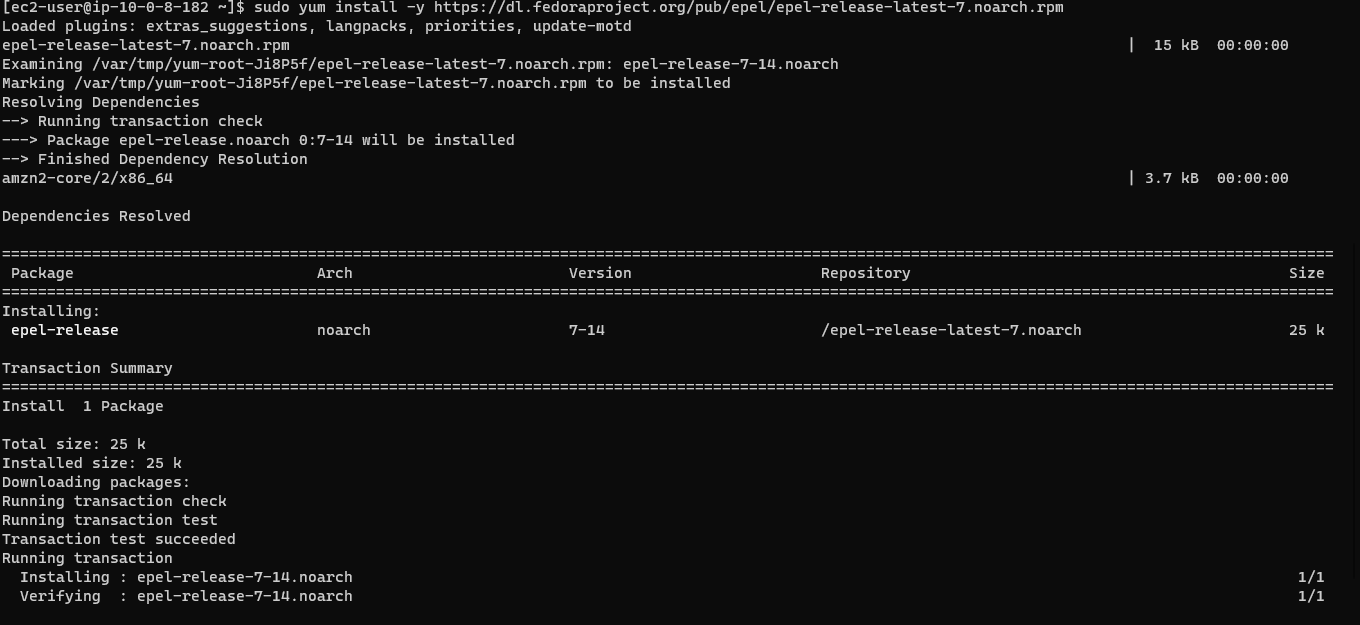
1. Install EPEL Repo
2. sudo passwd ec2-user
3. Install Google Authenticator on EC2 Instance
4. Configure EC2 SSH to use Google Authentication module
   1. Setup Google Authenticator module
   2. Configure Auth methods in SSHd config file
5. Configure Google Authenticator in Mobile & EC2
   1. Download and Install the GoogleAuthenticator in mobile
   2. Run Google Authenticator on EC2 and Get QR code
   3. Scan the Shown QR code in your Google Authenticator App
   4. Complete the Google Authenticator Setup in EC2
6. Restart SSH Services on the EC2 instance
7. Login to the server and validate
8. Video GUIDE of how to perform these steps

Step1: Install EPEL Repo on the EC2 instance

Google Authenticator is part of the EPEL repo and you should install the EPEL repo in your EC2 instance.

In my case, I am using Amazon EC2 Instance (OS) and I can download my EPEL 7 and install like this

*sudo yum install -y* [*https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm*](https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm)

**

* *Change hostname: sudo hostname jumpserver01*

**

Step2: Set Password For Ec2 instance (SSH) By Using This Command

# *sudo passwd ec2-user*

Step3: Install Google Authenticator on the EC2 instance

The following command would install the Google Authenticator. The package name may vary if your Operating system is different so you can perform a quick search using yum search google to find the right package name

*# sudo yum install google-authenticator.x86\_64 -y*

Step4:

Configure EC2 SSH to use Google Authentication module

In this step, we are going to perform some file modifications and editing to install google authentication and to setup MFA in this EC2 instance.

Step3a: Update the sshd PAM and install Google authenticator module

sudo vi /etc/pam.d/sshd

Add the following to the bottom of the file to use Google Authenticator. If there are service accounts or users who should be able to log in without MFA, add nullok at the end of the following statement. This will mean that users who don’t run Google Authenticator initialization won’t be asked for a second authentication.

auth required pam\_google\_authenticator.so or

auth required pam\_google\_authenticator.so nullok

Comment out the password requirement as we want to use only the key-based authentication.

#auth substack password-auth

Don’t forget to Save the file.

Step3b: Update the sshd configuration

In this step we are going to tell sshd that we have one more level of multifactor authentication for the user to login along with the Keybased auth.

This step is to make sshd daemon to prompt the user for the Verification Code.

Edit the file as root

#sudo vi /etc/ssh/sshd\_config

Comment out the line which says ChallengeResponseAuthentication ‘no’ and uncomment the line which says ‘yes’.

ChallengeResponseAuthentication yes

#ChallengeResponseAuthentication no

Finally, we need to let sshd daemon know that it should ask the user for an SSH key and a verification code

AuthenticationMethods publickey,keyboard-interactive

Save the file.

Step4: Configure Google Authenticator

It has two phases or two levels of configuration.