

Lab Setup on AWS

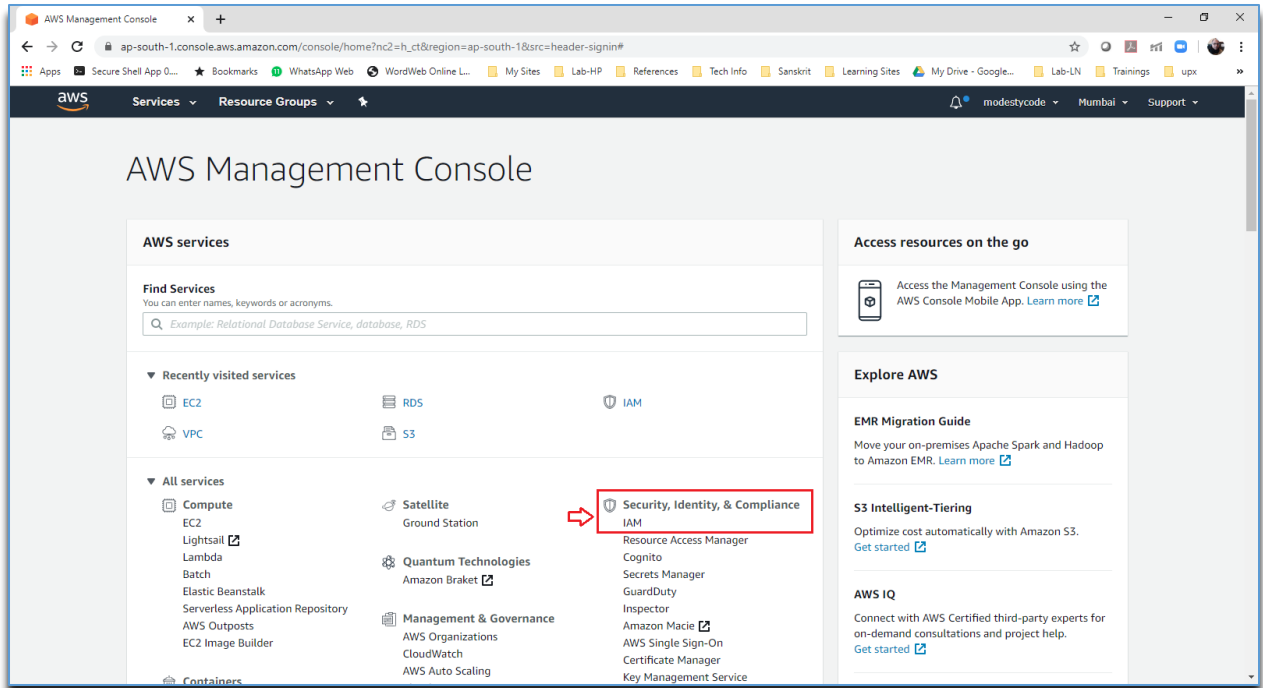
Time: 60 min

Create your AWS Account

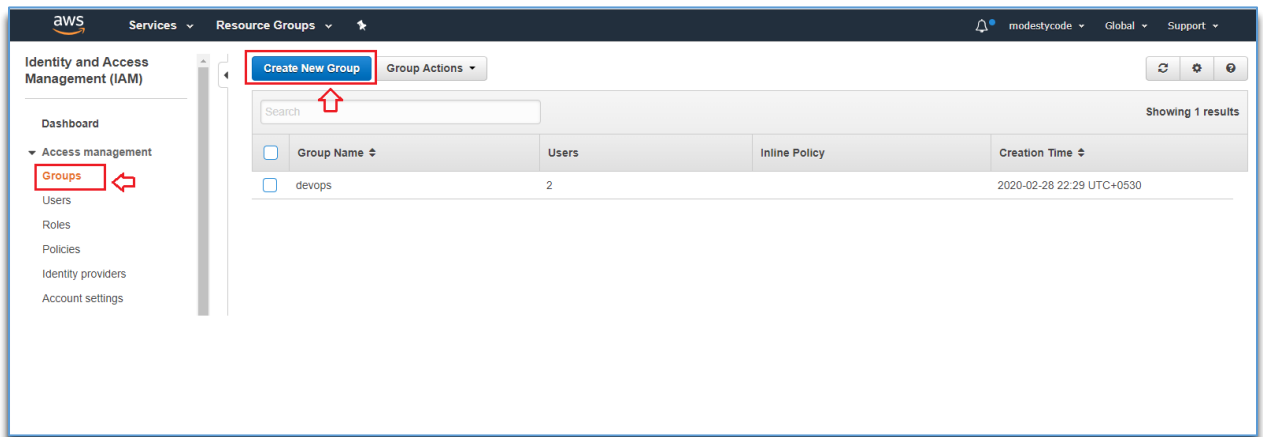
1. Go to the site as given below and follow the instructions
<https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/>
2. The account you created for the very first time is called a root account. Do not use it for day to day works – This account should be reserved for payments etc. In the next section we will create an administrator account and shall be using that only for all day-to-day admin works
3. Go to next section now

Secure your AWS account

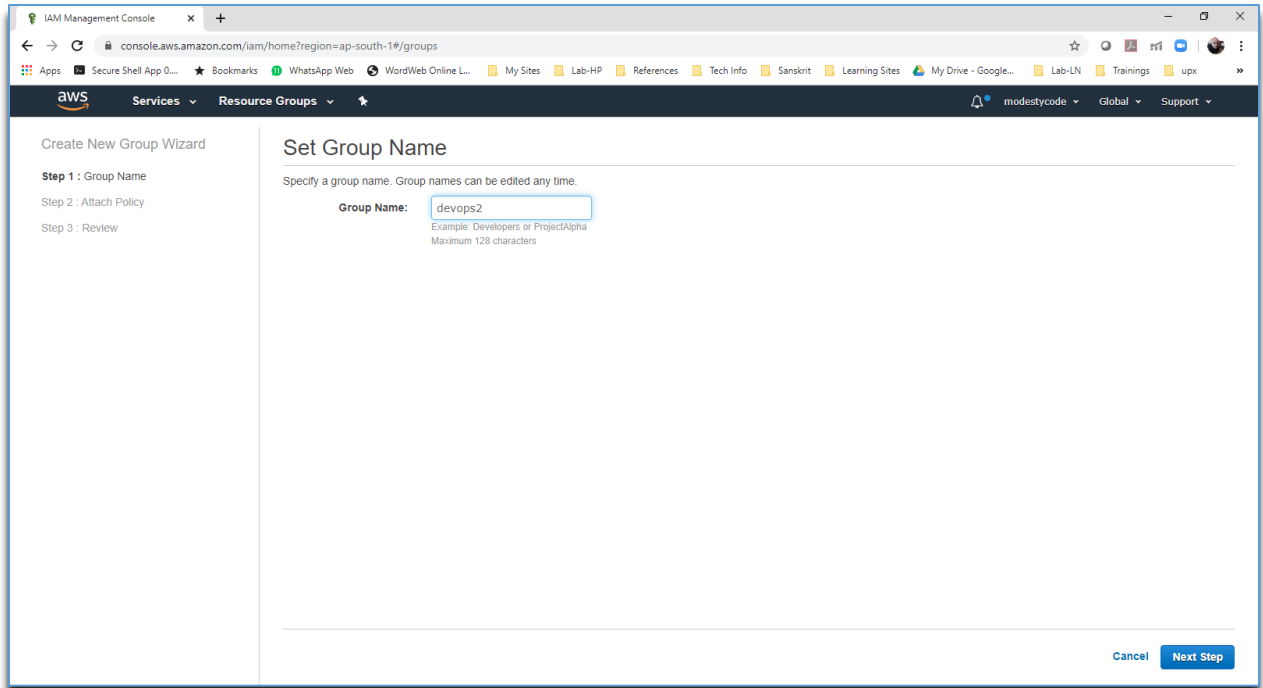
1. Having created your root account, now we will create an administrator account in this section. Also, we will apply some very basic and essential security features to your account
2. Login to your AWS account with root user email and password – use the username and password you have created when you created the account in previous section
3. In the landing page, click on **IAM** service under **Security, Identity and Compliance**



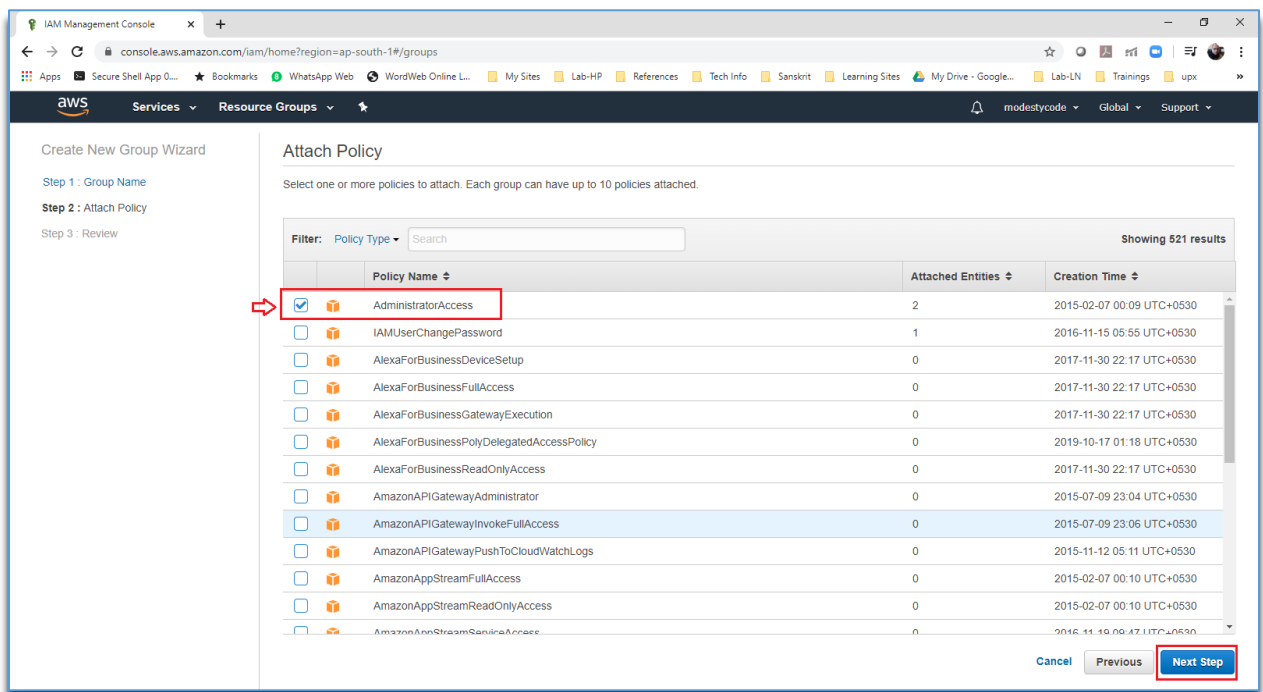
4. Create a **Group** first, we will assign user to the group later. Click on the Group menu on left panel. Ignore the security alerts shown in the main page – This will be all Ok once we finish the tasks in this section
5. Click on **Create New Group** button at the top in the Group Management group



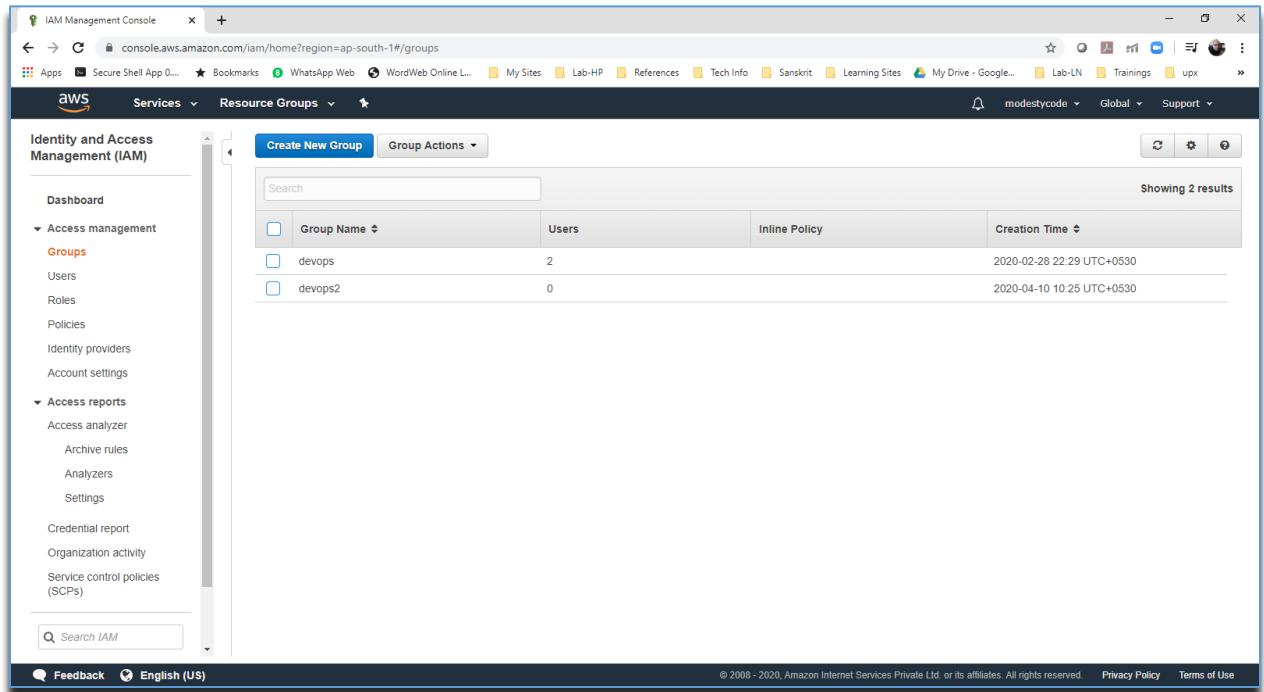
6. Provide a group name and click on **Next** button at the bottom right corner of the page



- Now we have to attach a policy to this group. This will set the selected access and permission whoever belong to this group. Select **AdministratorAccess** for the list and click **Next Step** button at the bottom right

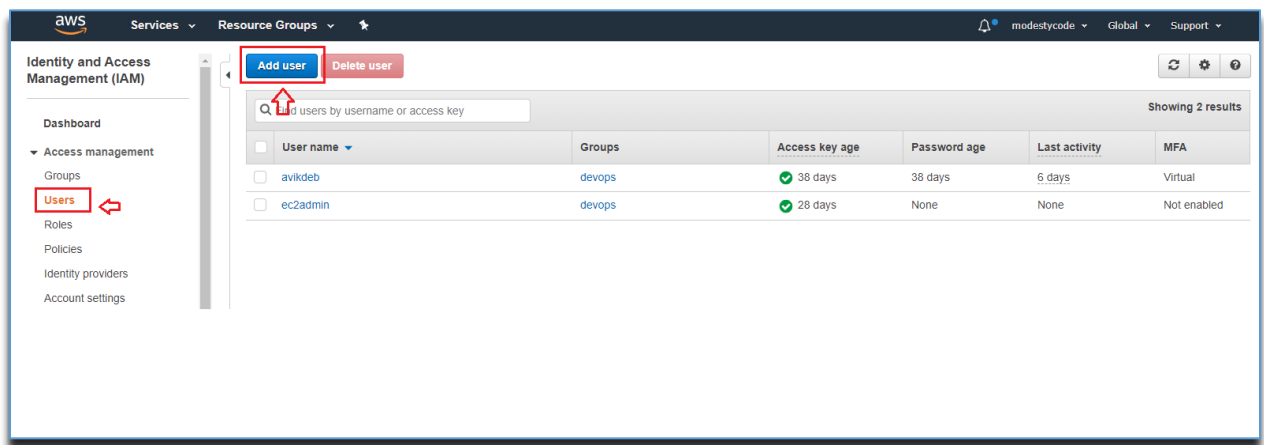


8. In review screen click **Create Group** button at the bottom right. New Group is now created and visible.

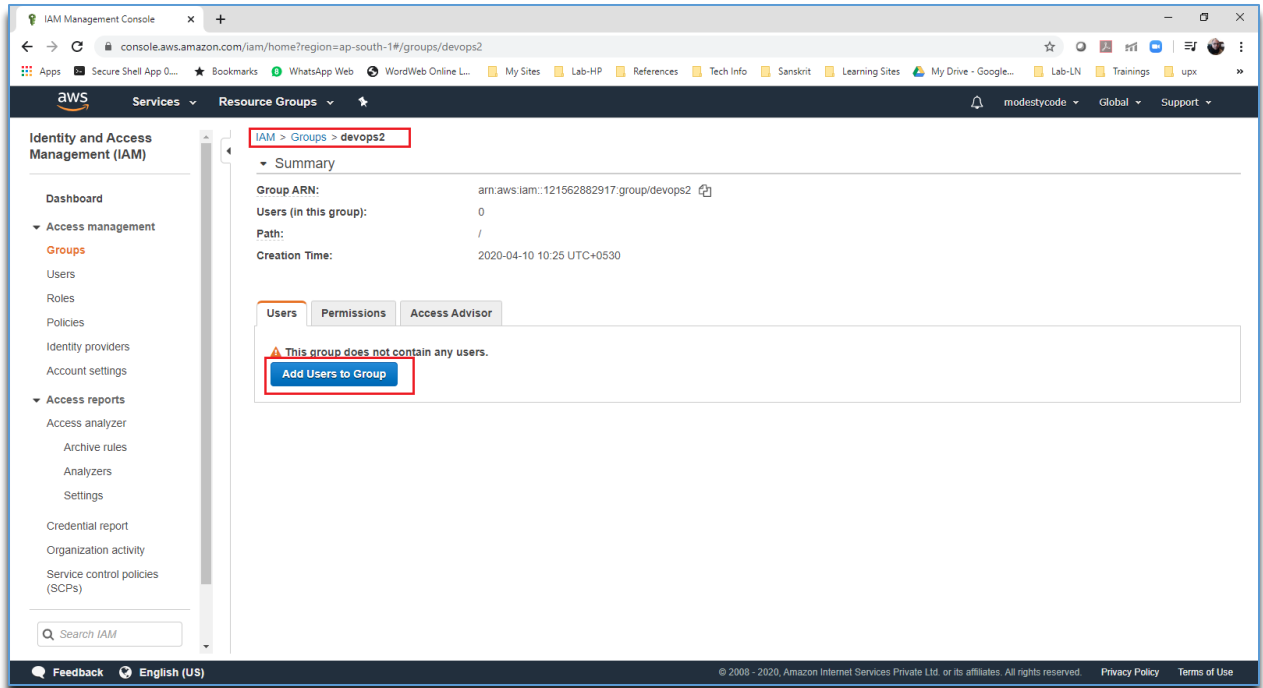


9. Proceed to create users and add them to this group. They will inherit the permission set at the group level
10. Click on the **User** menu on left to create your first IAM user – This user will be used for all day to day admin works.
11. Click on **Add User** button at the top in the User Management screen (or, you can navigate from the selected **Group** screen > **Add Users to Group**)

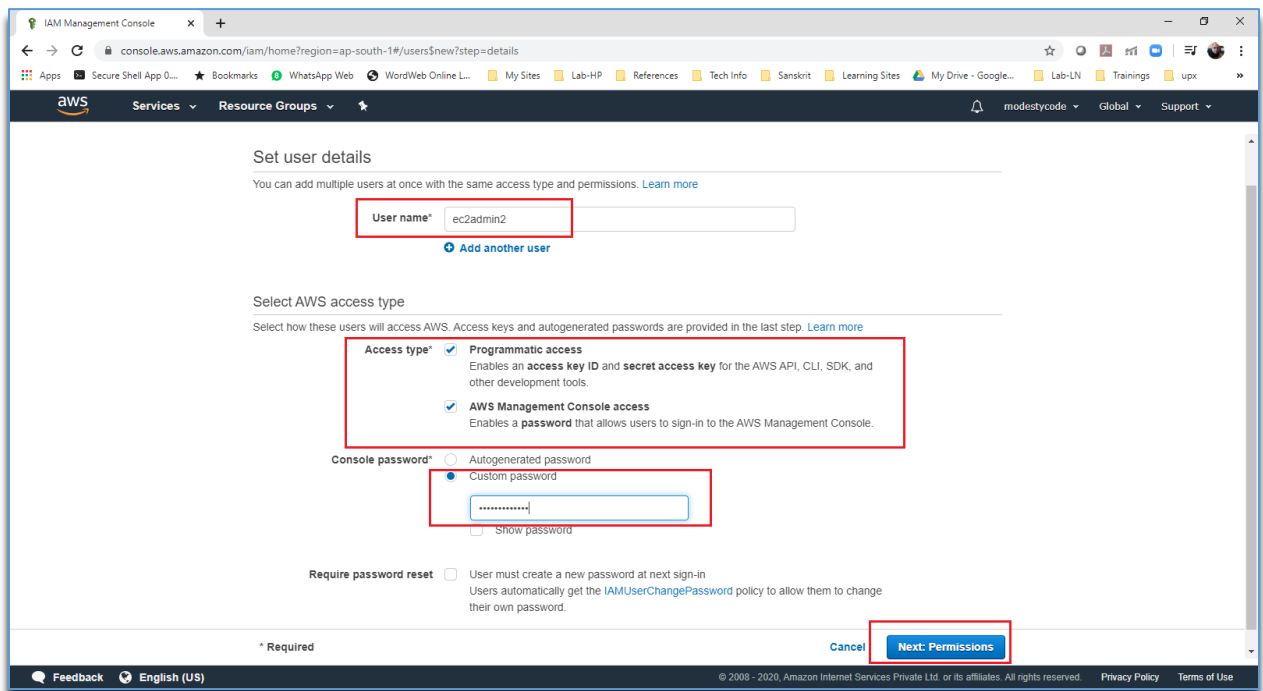
From User Menu:



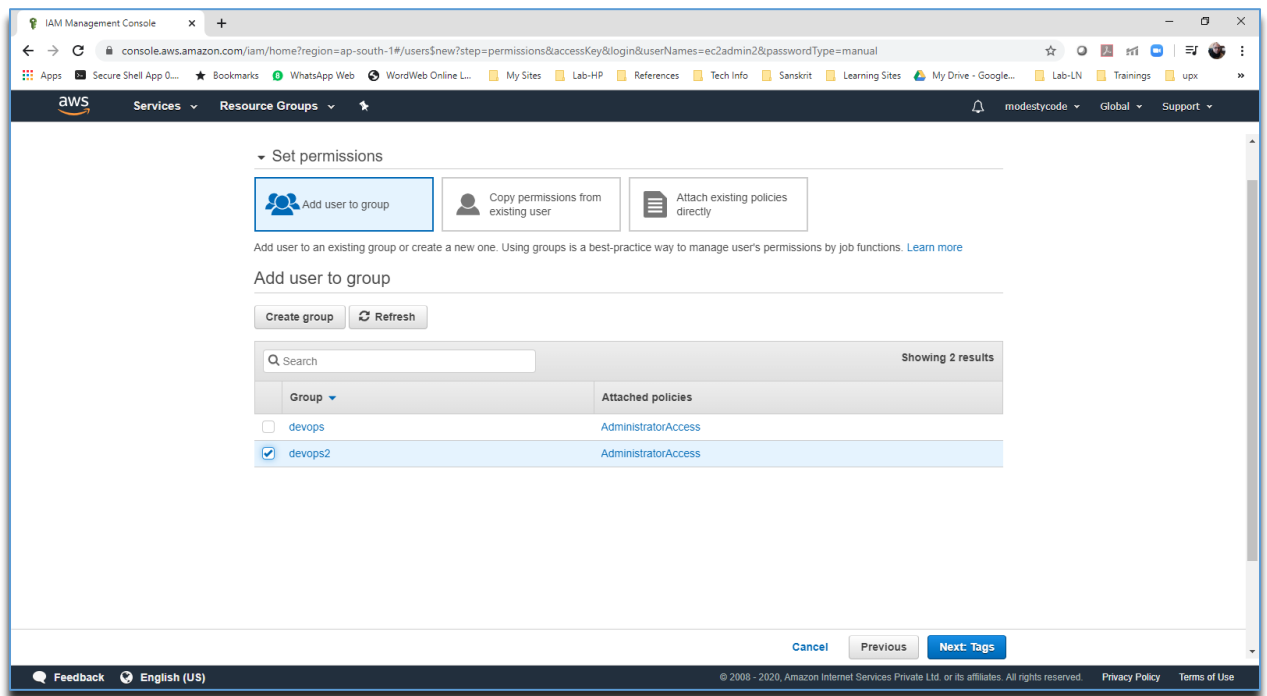
From Group Screen (evoked from the selected group):



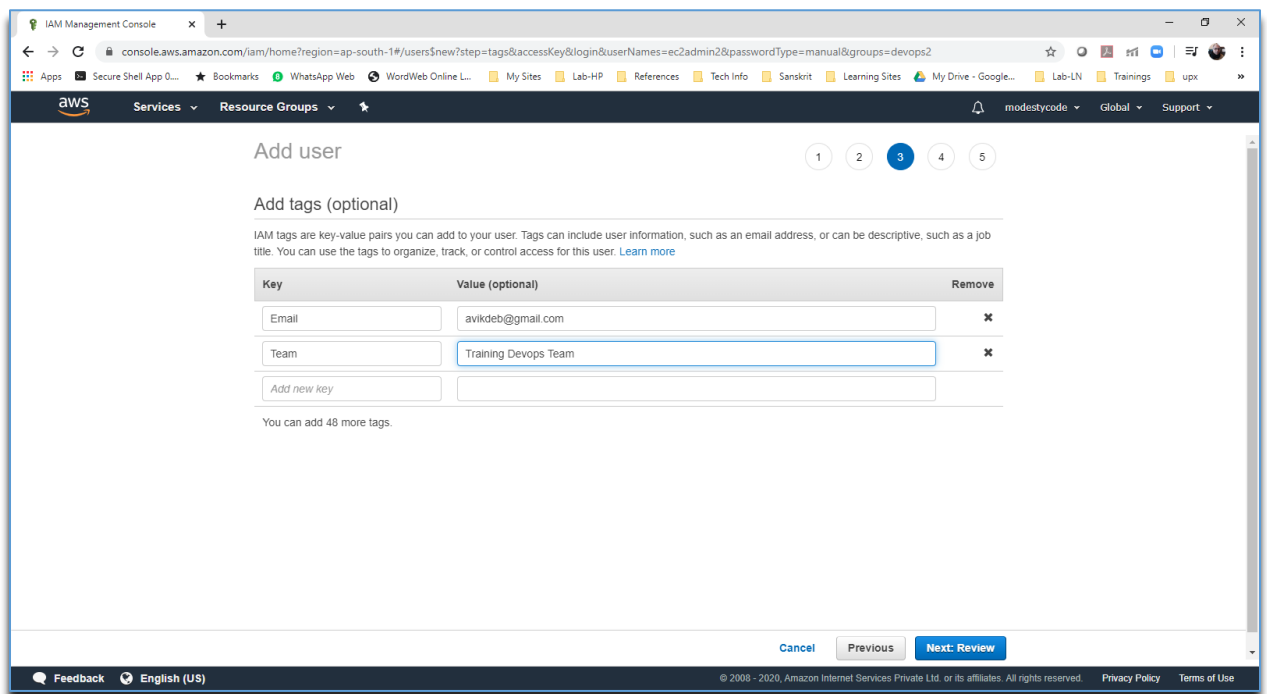
12. Set the user details as follows – Give this user both console access (as you will be using it for all day-to-day operations) and programmatic access (later you may use for automation). Click on **Next: Permissions** button at the bottom



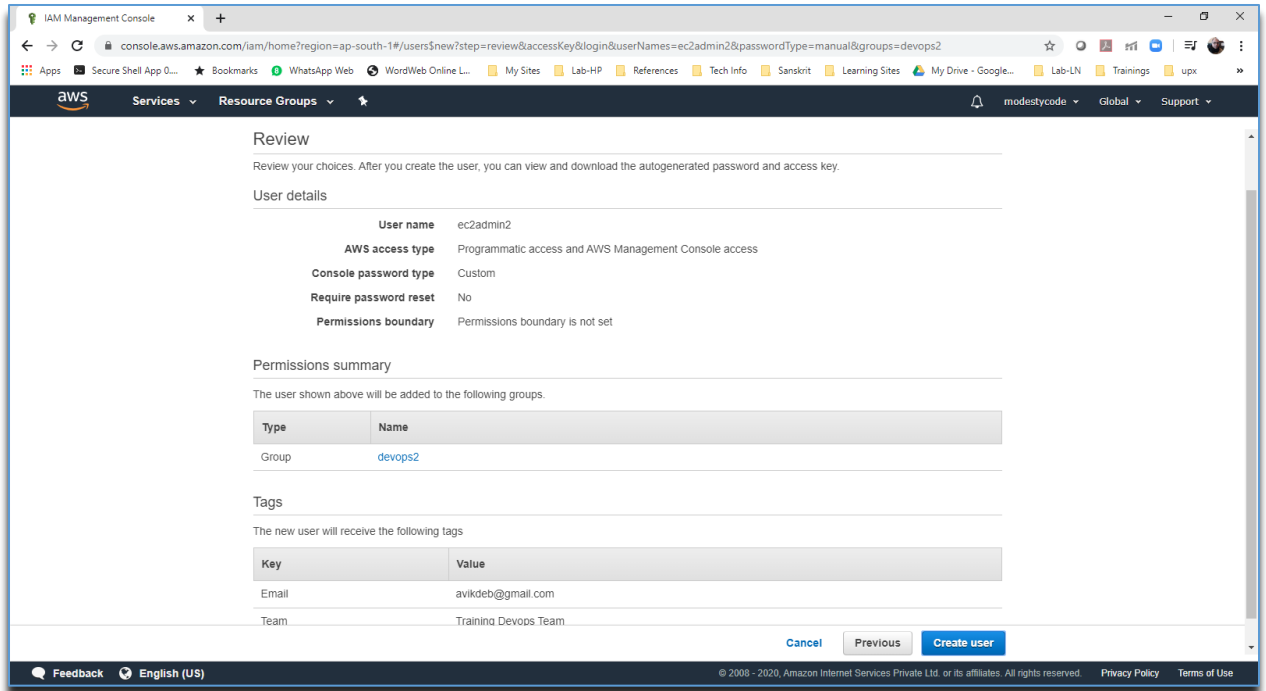
13. Now add this newly created user to the chosen group – we have already created the group with desired access set. Click on **Next: Tags** button at the bottom right



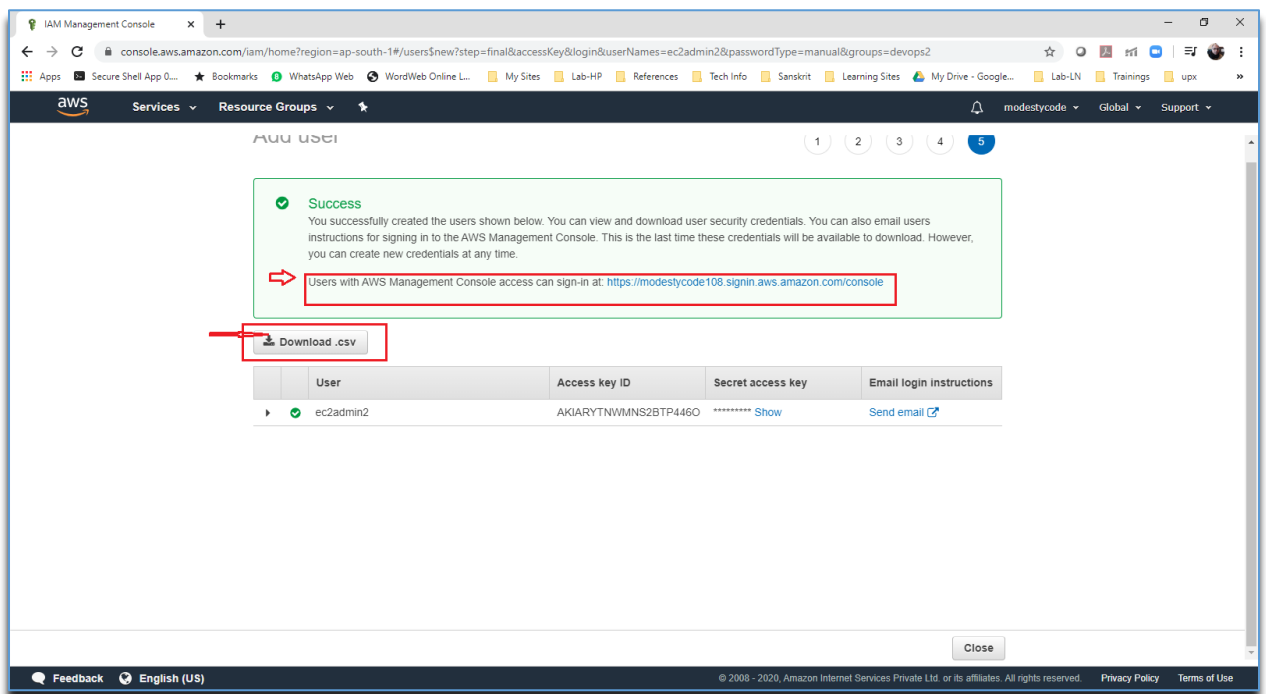
14. Provide some useful and relevant information about this user as key-value pair in Tags screen.
Click **Next: Review** button on bottom right



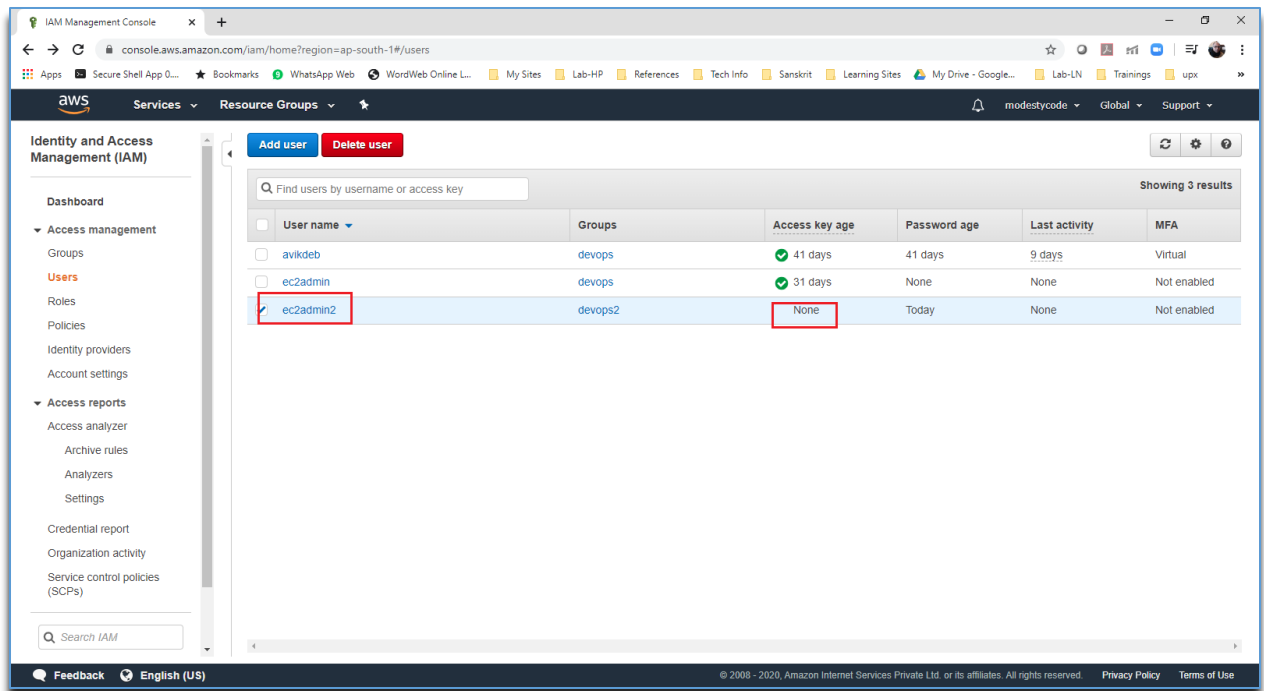
15. In the review screen, click on Create user button at the bottom right



16. User is created – download the .csv file containing relevant information, will be required for programmatic access in future. Note the URL – This will be used for console login. Click **Close** button

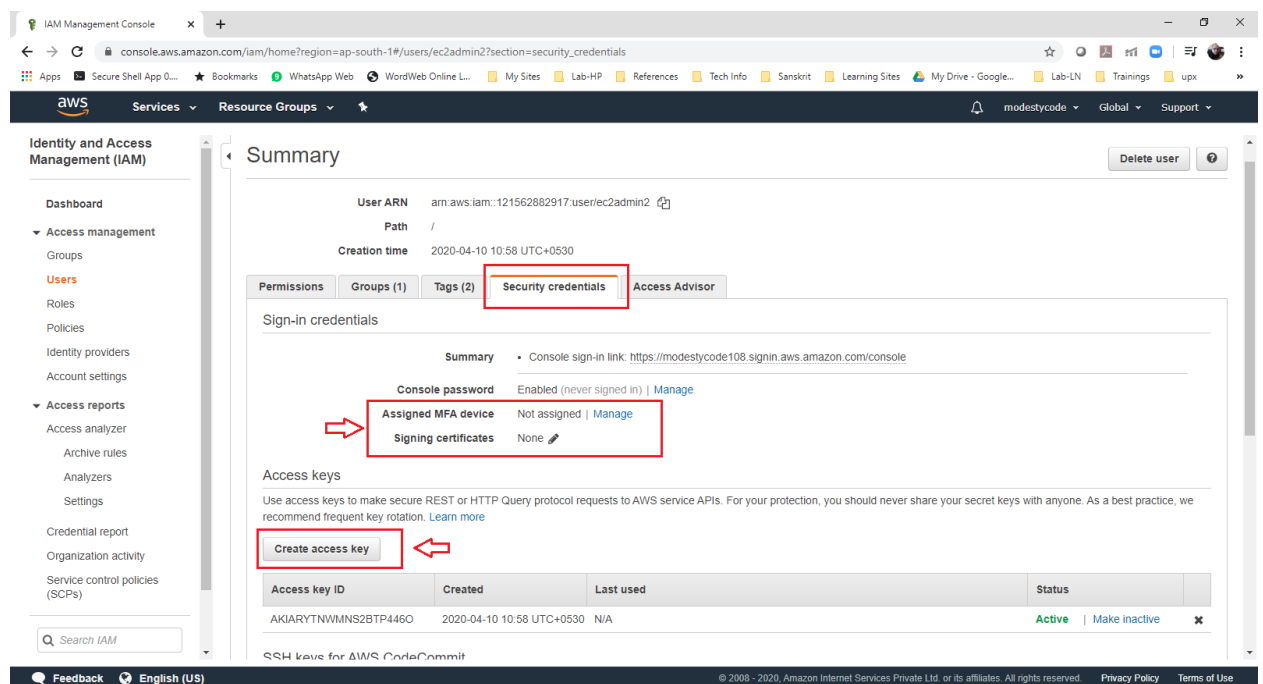


17. Now we need to modify access key etc. for this user. Select the user in list to modify

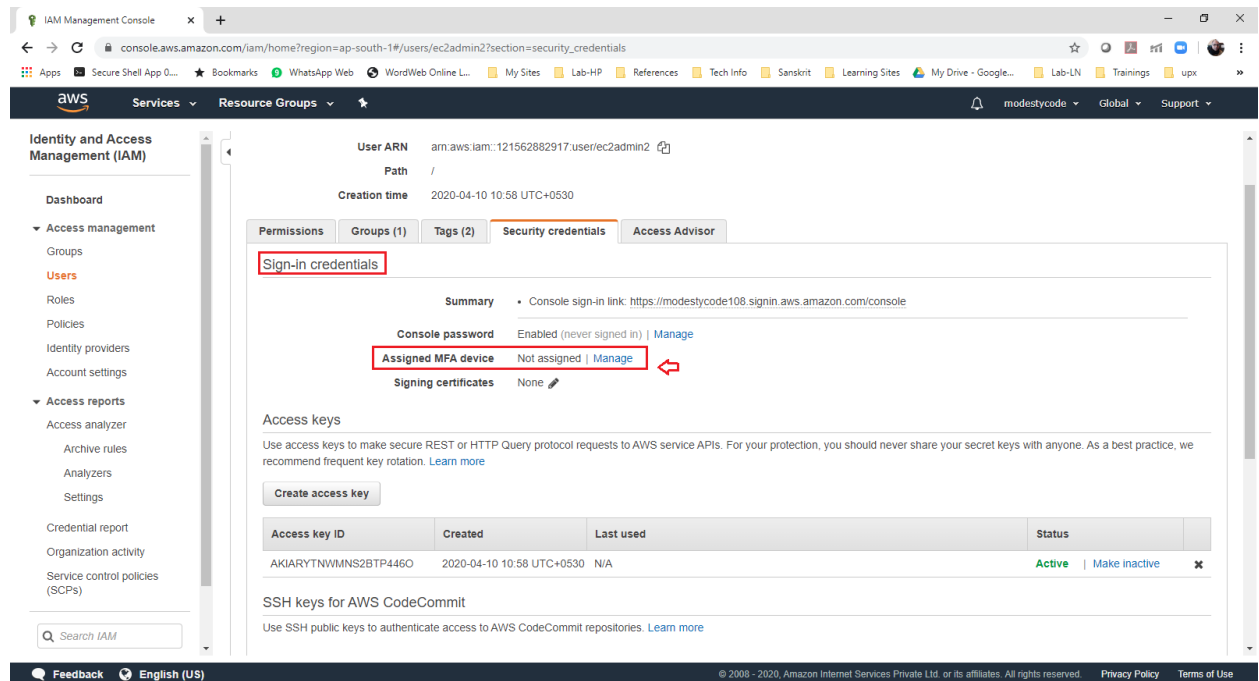


18. Click on Security Credentials tab. We need to enable MFA.

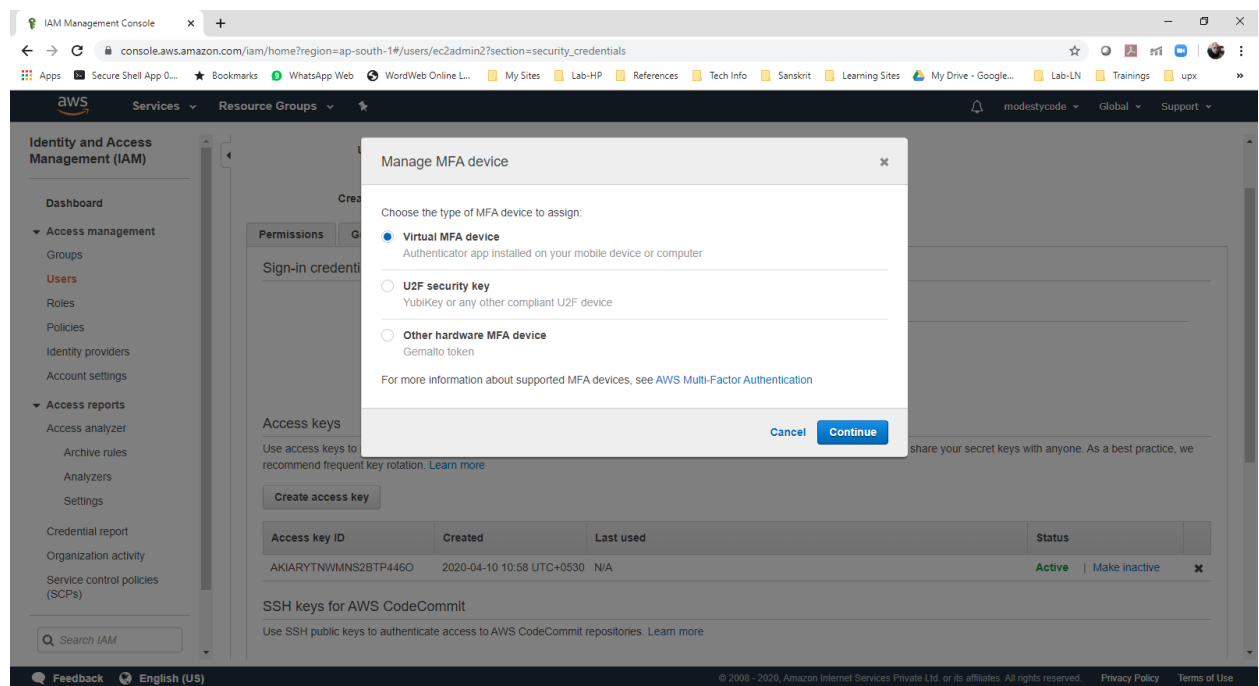
Enable MFA – This required Google Authenticator apps to be downloaded to your mobile phone. This will generate 2-Factor Authentication keys required for login to the console (after providing username and password, you will be asked to provide MFA key. Open google authenticator in your mobile and get the MFA key and type-in the same to access the console)



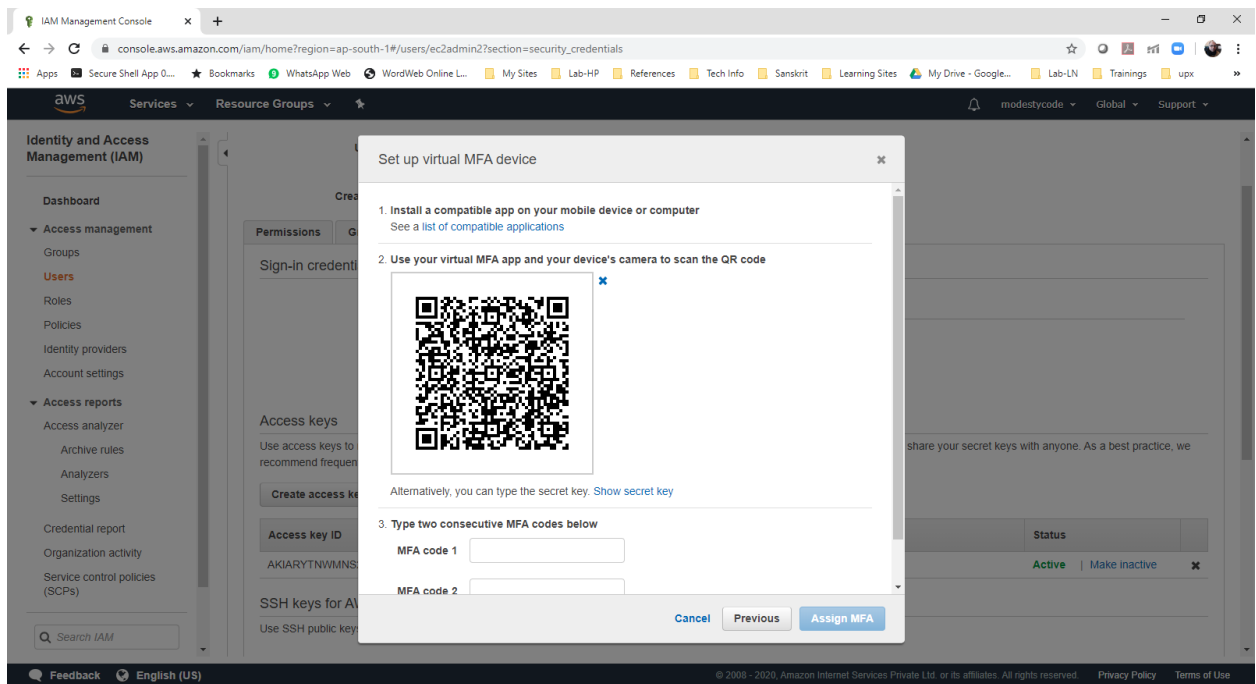
19. For MFA: Click on **Sign-in Credentials** > **Assigned MFA device** > **Manage**



Select **Virtual MFA device** – This points to your Google Authenticator which you have downloaded from Google Play Store (for Android) or Apple App Store (for iPhone)



Click on **Show QR code** – Open your Authenticator App in mobile and scan the code shown here

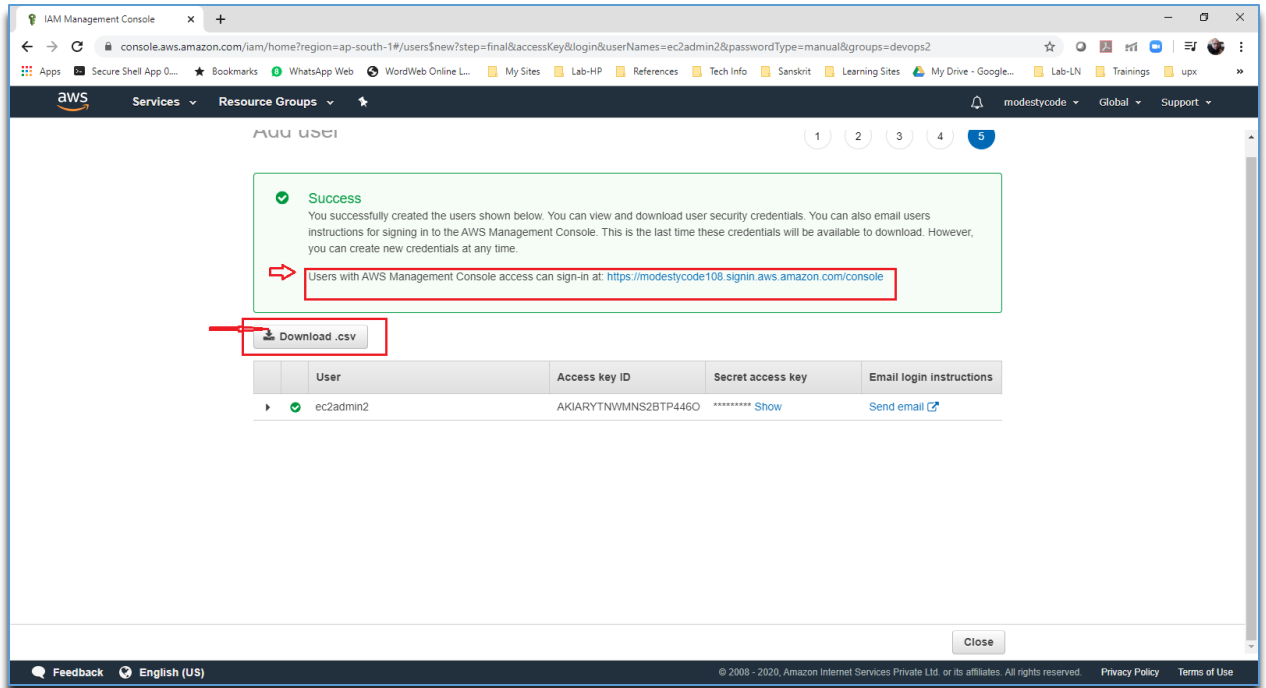


- Type the two consecutive MFA codes as instructed – You must wait and get 2 MFA code. Allow some time to change the codes in your authenticator app in mobile. Finally click **Assign MFA**
20. Now you can log in to AWS console with this user and use MFA code for extra bit of security
21. This completes the IAM tasks

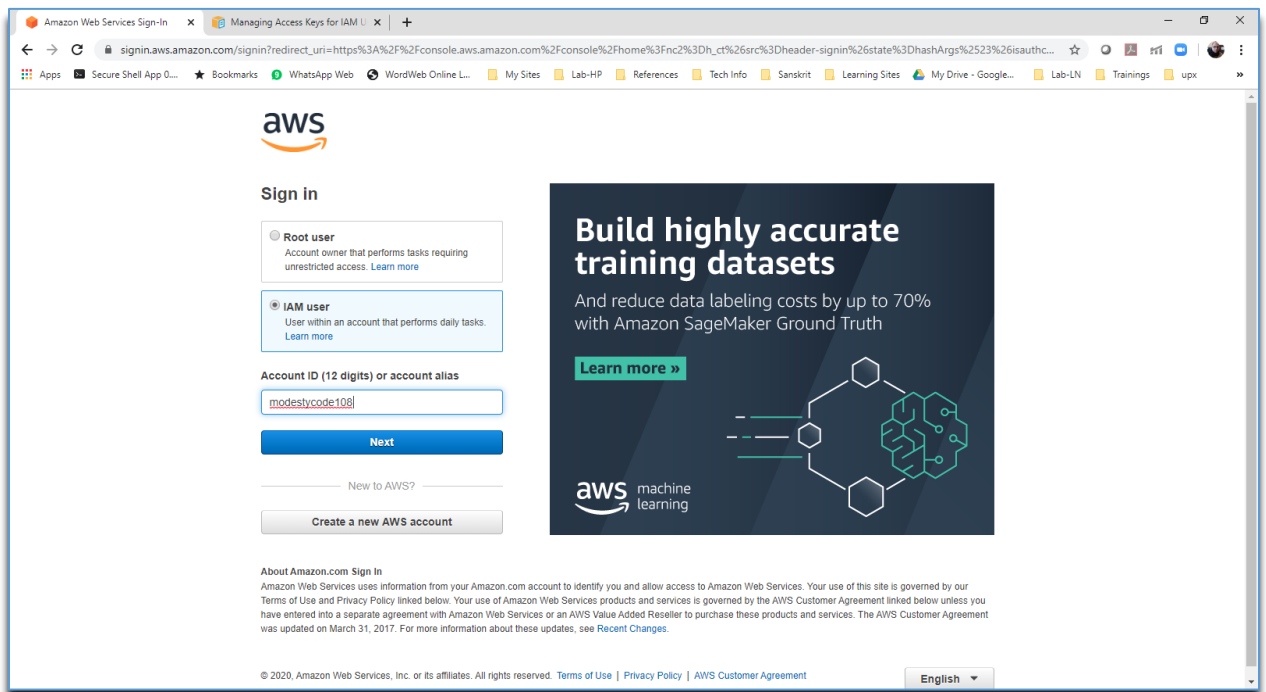
Create your EC2 instances

1. Login to AWS console with IAM user created in the previous section – use the MFA code generated in the Google Authenticator Apps on your mobile phone, when asked during the login. Select the **IAM user** this time and **NOT** the root user. Provide the account ID you have saved during account creation – point number 16 in above section (see the screenshot below you have seen in previous section). The first piece after https:// (in the example, modestycode108 is the alias set when we created the account. You will have yours own) is the account ID or alias (if you have set).

If not, you can log in as root user one more time and get that)



Login Screen:



2. Provide your IAM username and password and then provide MFA code. You are in AWS services page or welcome page. Select EC2

Amazon Web Services Sign-In x Managing Access Keys for IAM L x +

sign-in.aws.amazon.com/oauth?redirect_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3Fnc2%3Dh_ct%26src%3Dheader-signin%26state%3DhashArgs%2523%26isauthc...

aws

Sign in as IAM user

Account ID (12 digits) or account alias

modestycode108

IAM user name

avikdeb


Password

.....

Sign in

Sign in using root user email

[Forgot password?](#)



Amazon RDS Multi-AZ

Enable RDS Multi-AZ configurations for a more resilient DR Strategy

English

[Terms of Use](#) [Privacy Policy](#) © 1996-2020, Amazon Web Services, Inc. or its affiliates.

Amazon Web Services Sign-In x Managing Access Keys for IAM L x +

sign-in.aws.amazon.com/oauth?redirect_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3Fnc2%3Dh_ct%26src%3Dheader-signin%26state%3DhashArgs%2523%26isauthc...

aws

Multi-factor Authentication

Enter an MFA code to complete sign-in.

MFA Code:

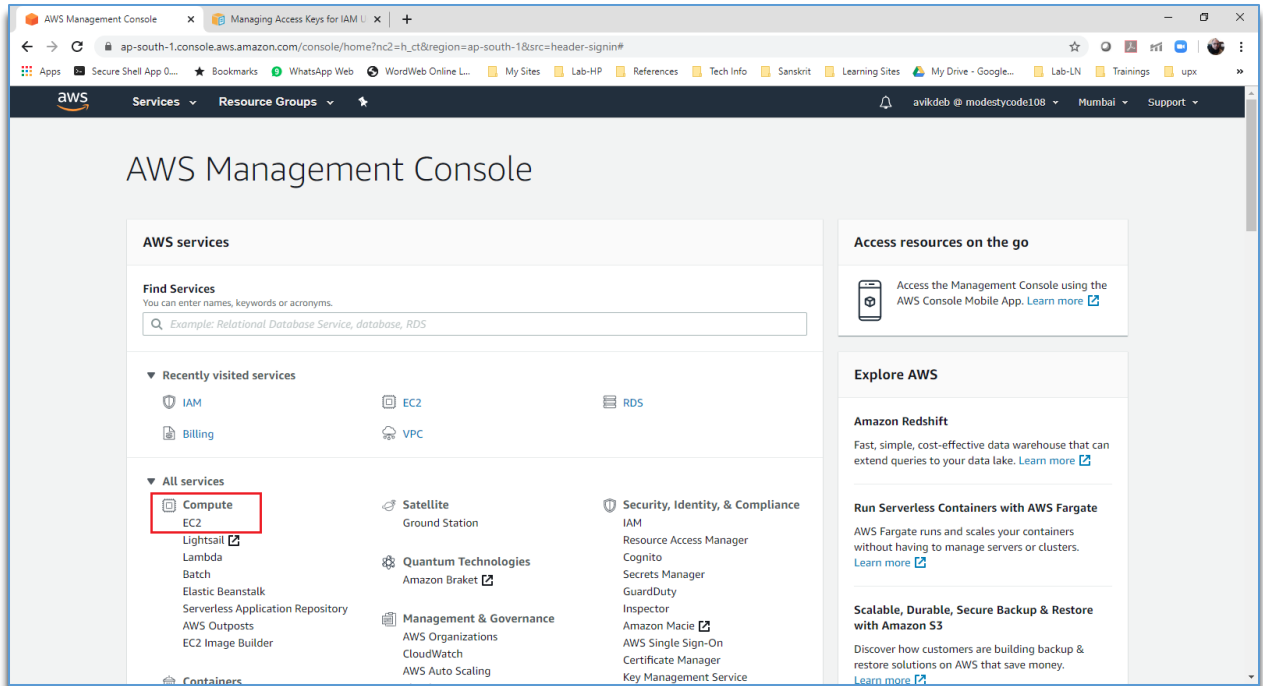
128906

Submit

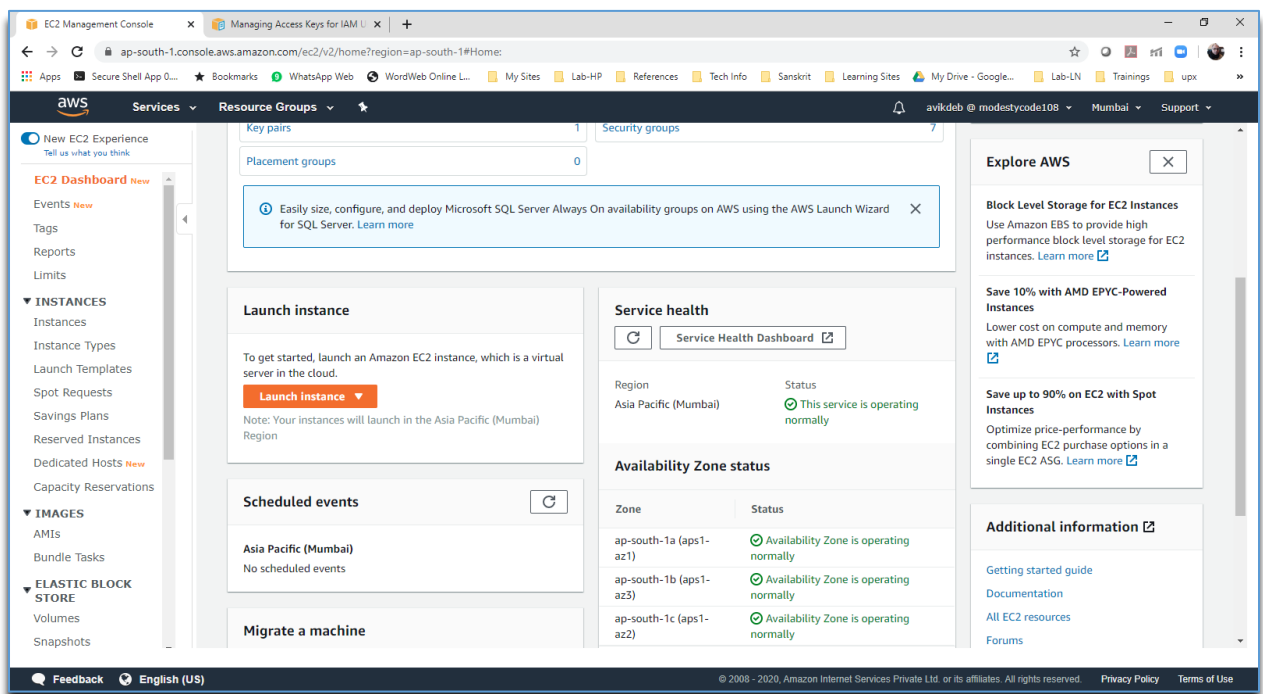
[Cancel](#)

English

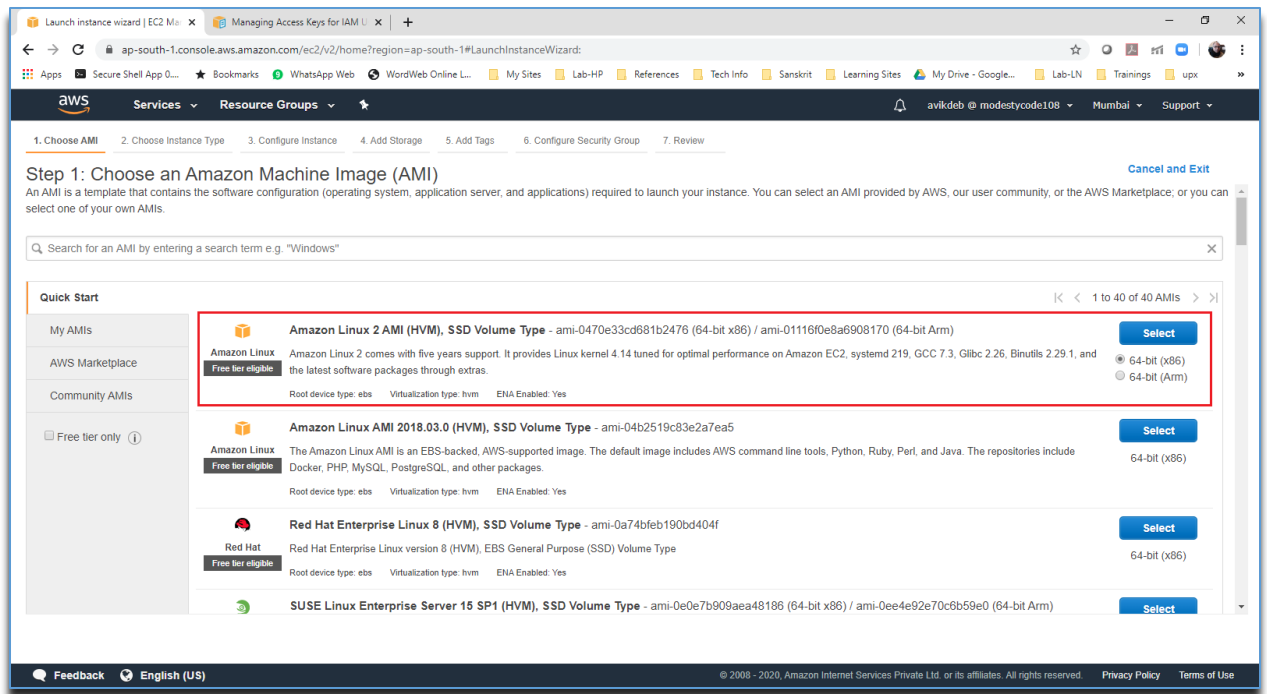
[Terms of Use](#) [Privacy Policy](#) © 1996-2020, Amazon Web Services, Inc. or its affiliates.



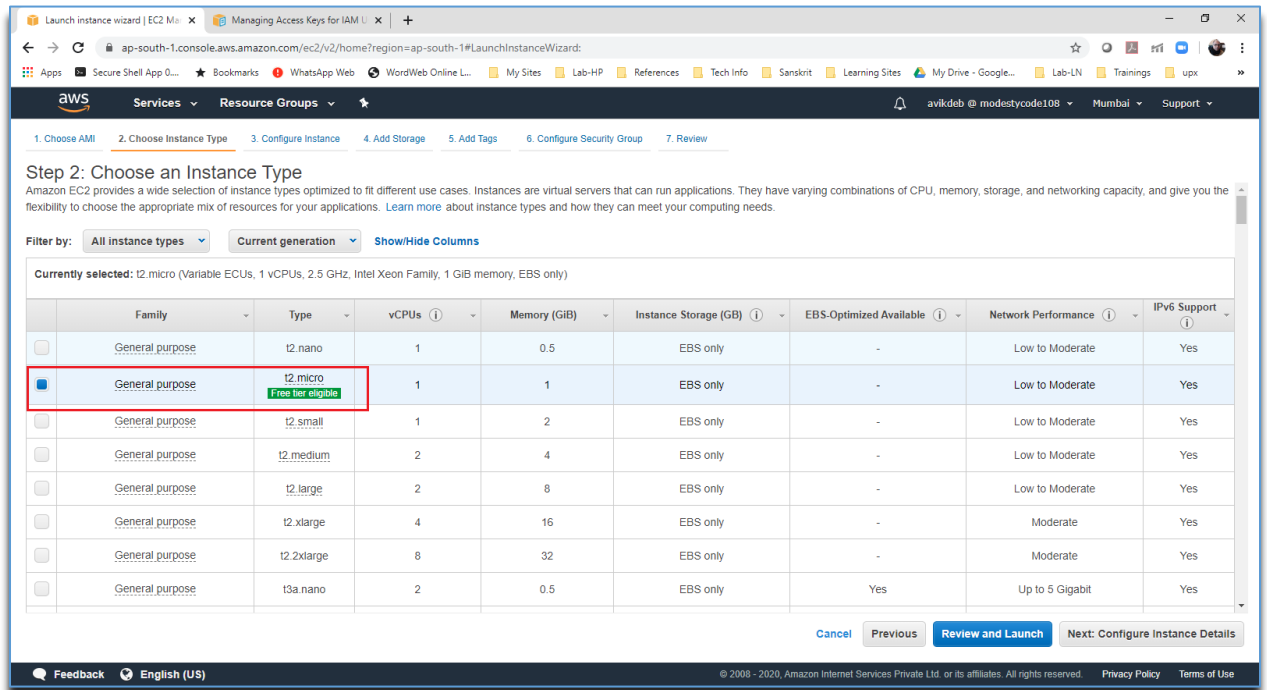
3. Select EC2 and Then Launch Instance



4. Select the appropriate image as shown below and click on **Select**

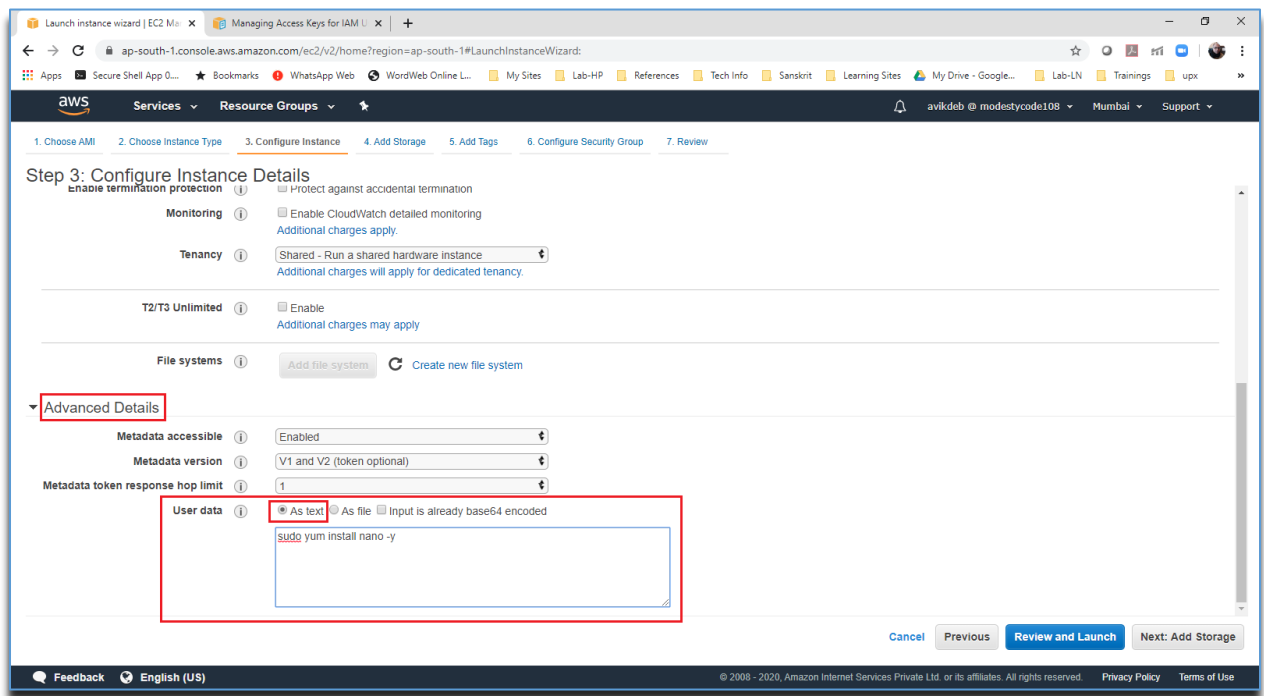


5. Select the Family as **t2.micro** – Check your selected family should be **Free tier eligible**. Click **Next: Configure Instance Detail** button on the bottom right



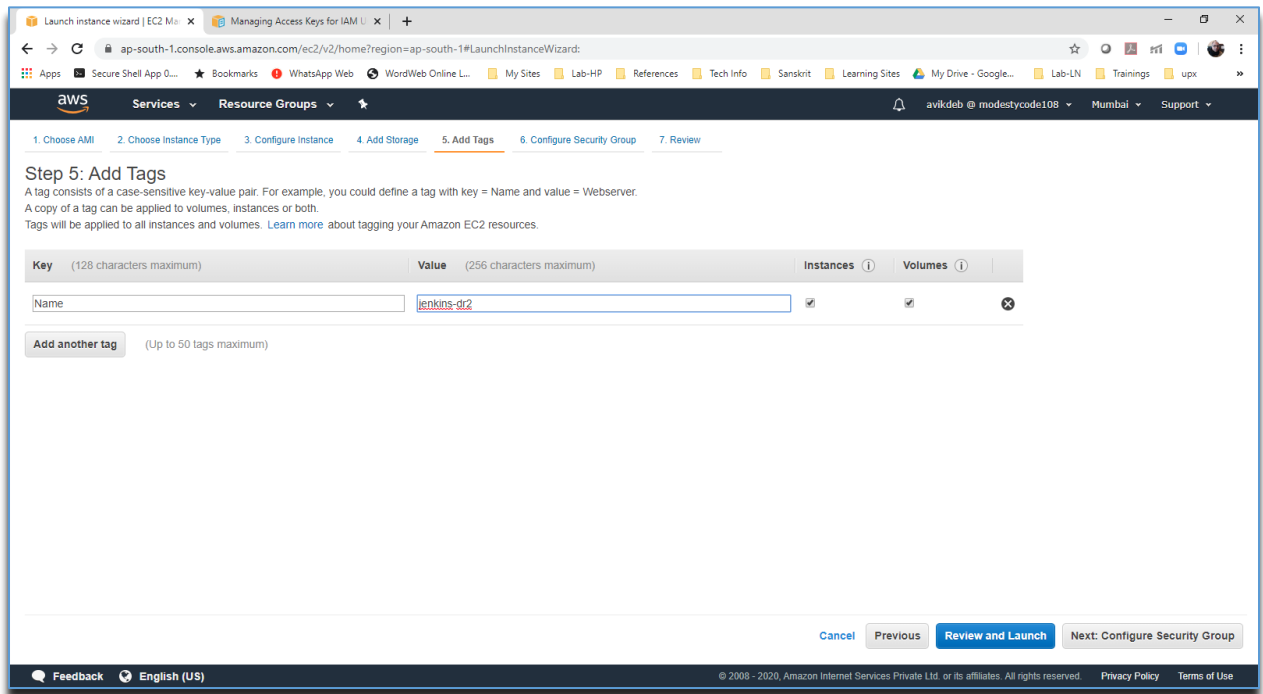
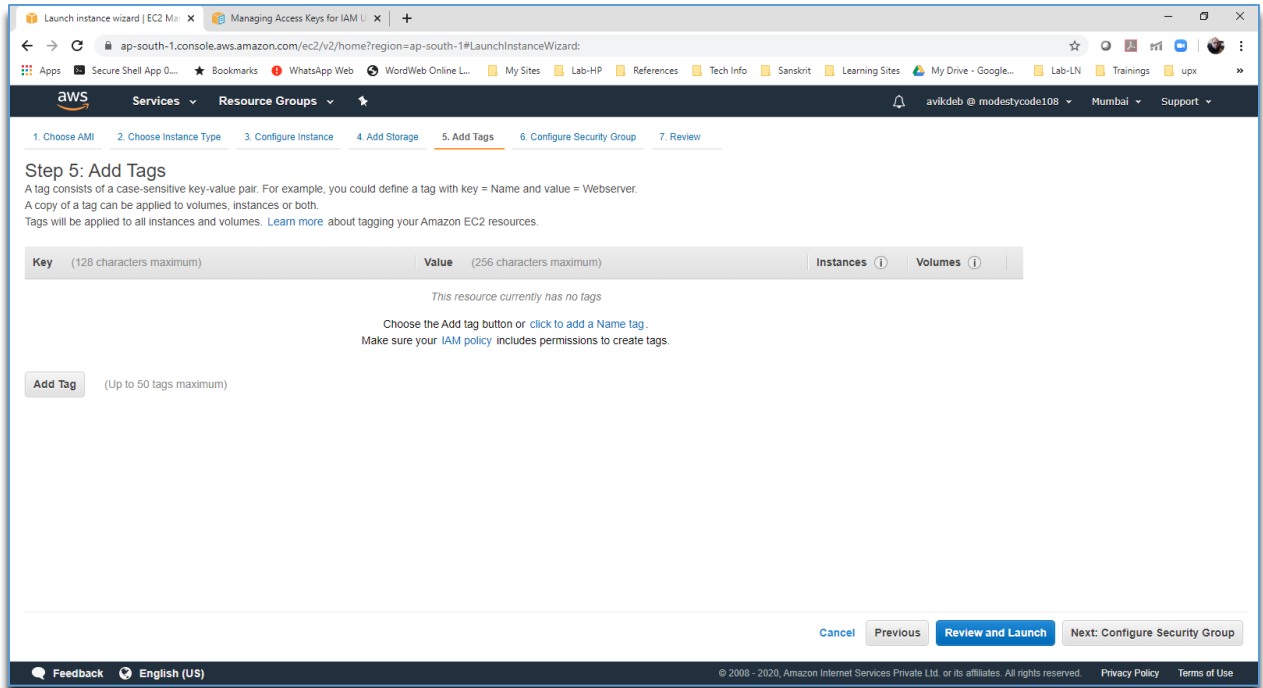
6. Keep all default option as-in in the Instance detail page. However, you can run a bootstrap script in case you need anything to install etc. during provisioning. Let us install nano during

bootstrapping. Scroll down to **User data** section under **Advanced Details** and write steps for nano install as given below:



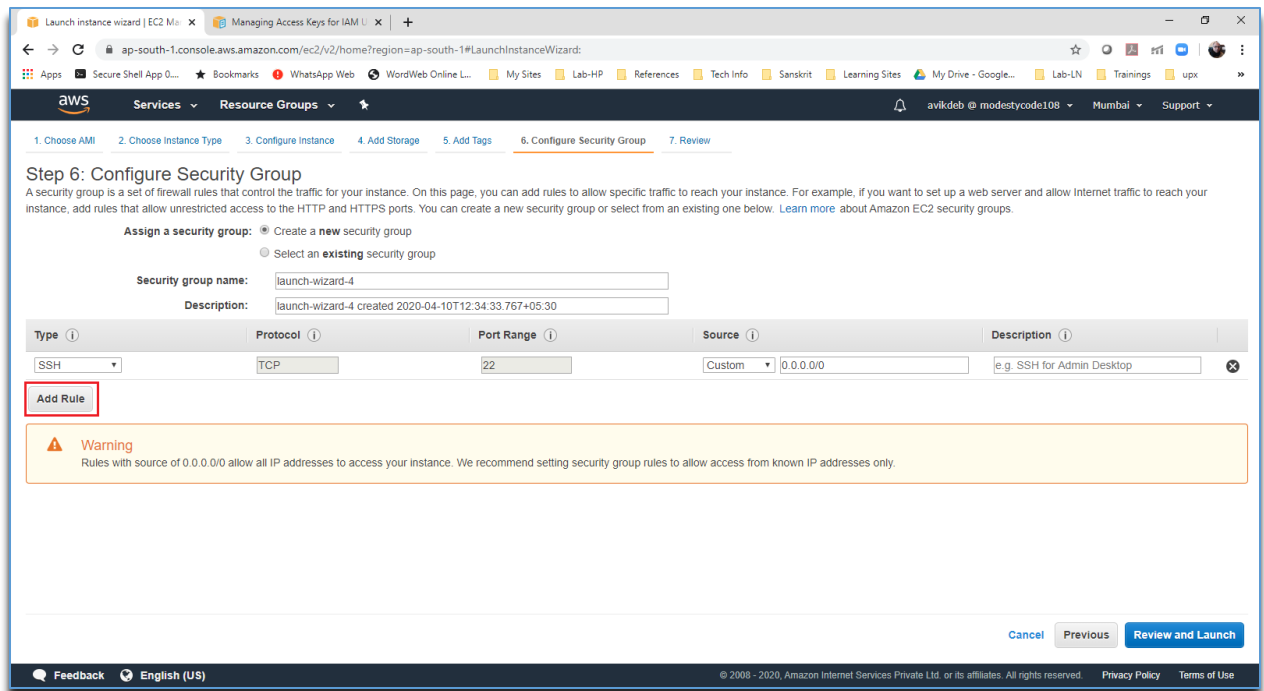
Click on **Next: Add Storage** button at bottom right

7. Leave the default settings as-is. Default space allotted to your instance is 8GB which is good enough for all practical purposes. We can anytime increase this space. Proceed to **Next: Add Tags**
8. Provide a name of your instance as key-value pair. Click on Add Tag button and provide Name Value pair

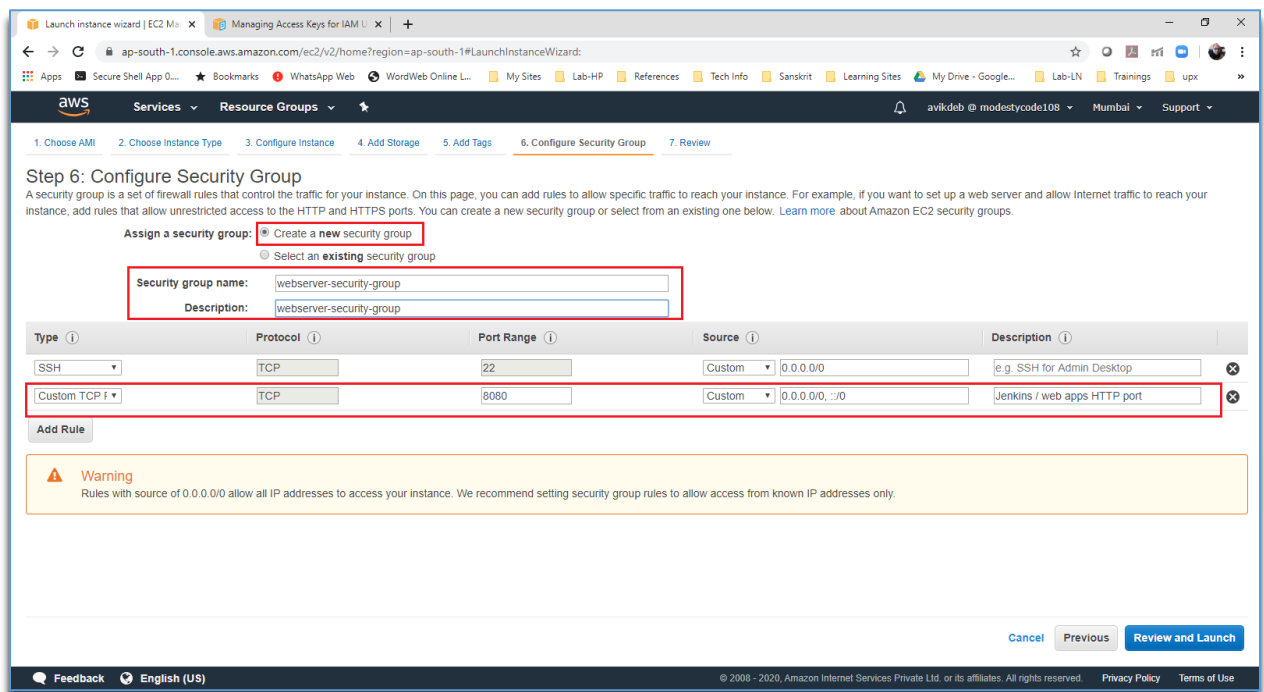


Click on **Next: Configure Security Group** button at the bottom right

- Define a security group for the instance – and we will be using this security group settings for all similar instances. Let's name it as webserver-security-group. In this step, we are actually configuring the firewall and we are telling which ports to remain all to whom. As Jenkins runs on 8080 by default, we will open all inbounds traffics on port 8080. Click on **Add Rule** button

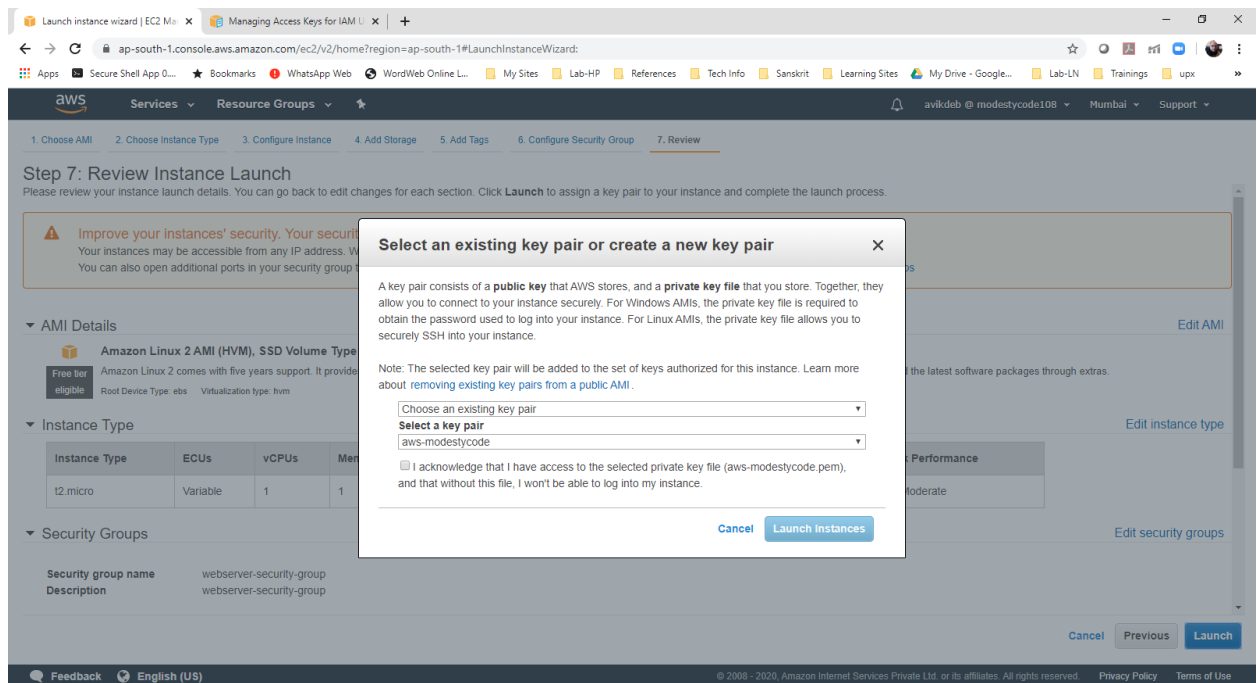


10. Give name and description of this security group as **webserver-security-group** instead of the default name provided. Select Custom TCP/IP and open port 8080 for all networks. Click on **Review and Launch** button at the bottom right



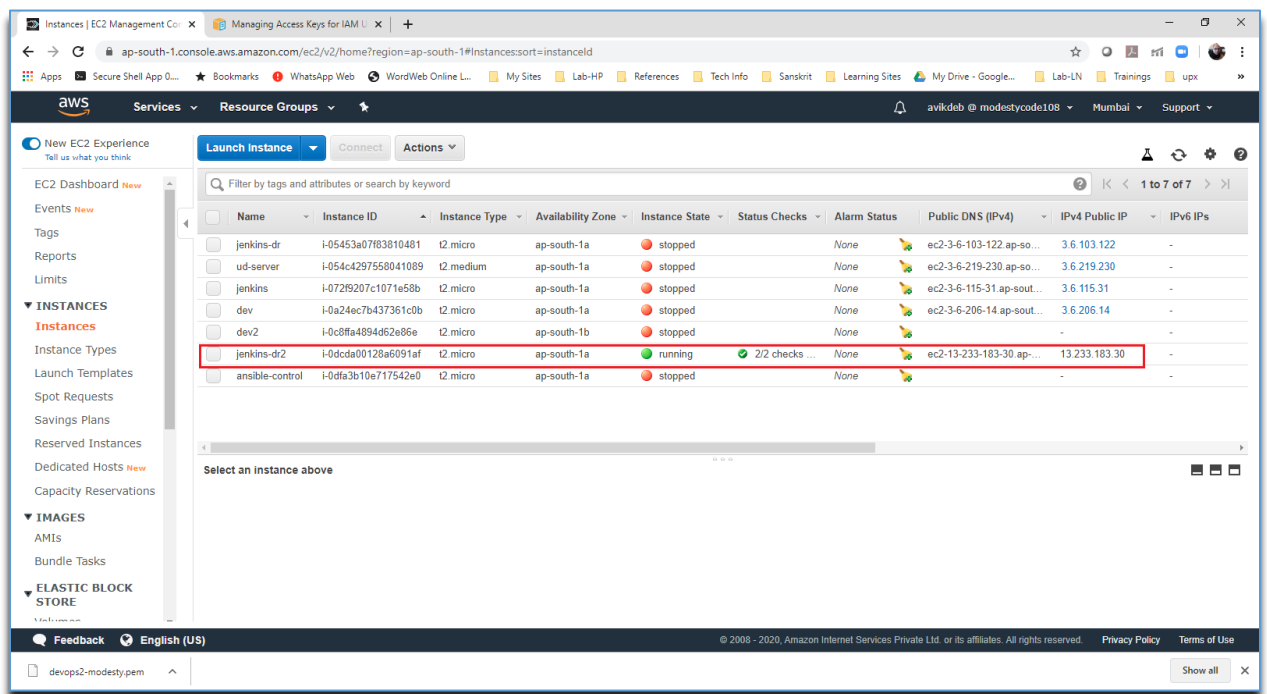
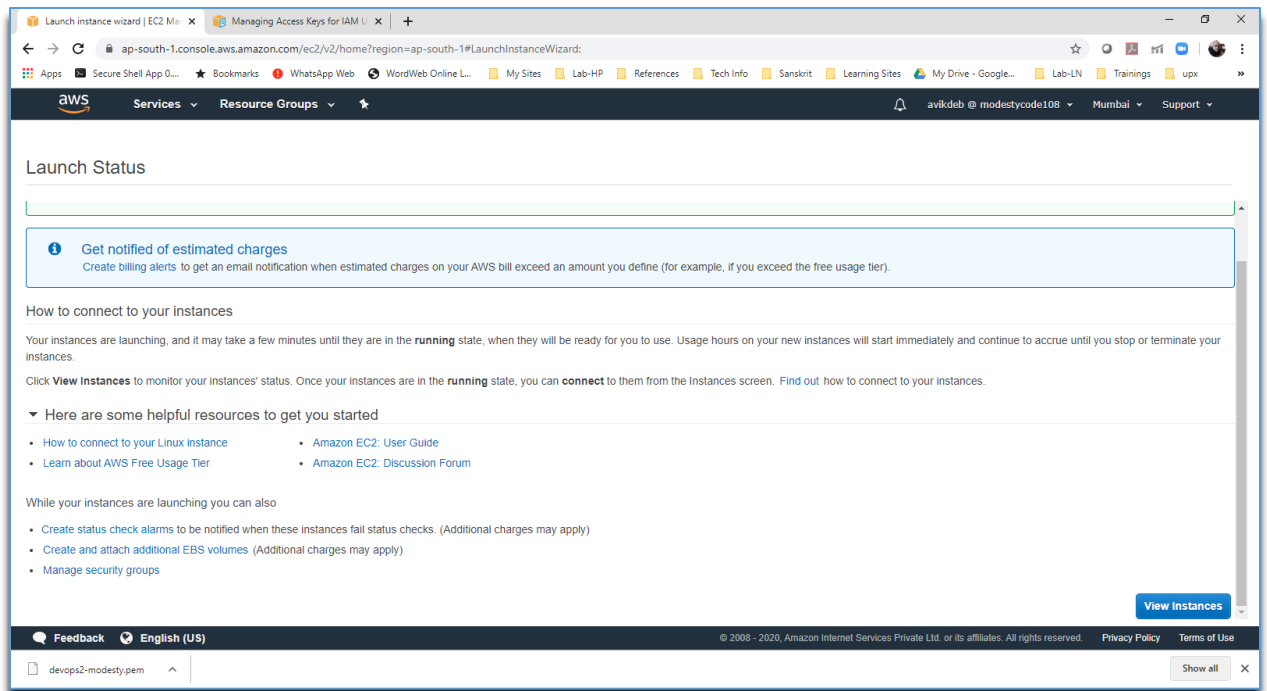
11. Review the information provided and click on **Launch** button at the bottom right

12. Select new key-pair and give a meaningful name. This will be required to remotely login to the box. Download and keep this safe. It gives you an asymmetric key in .pem file.

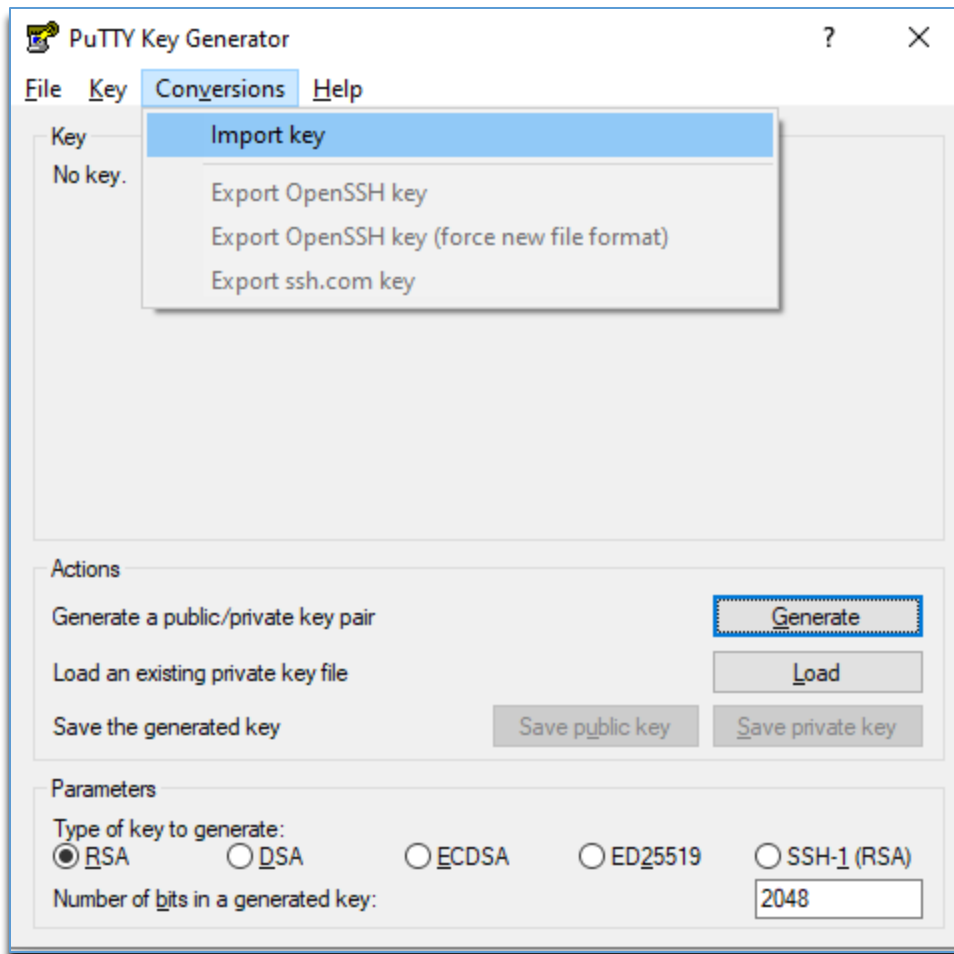


Click on **Launch Instances** button on bottom right

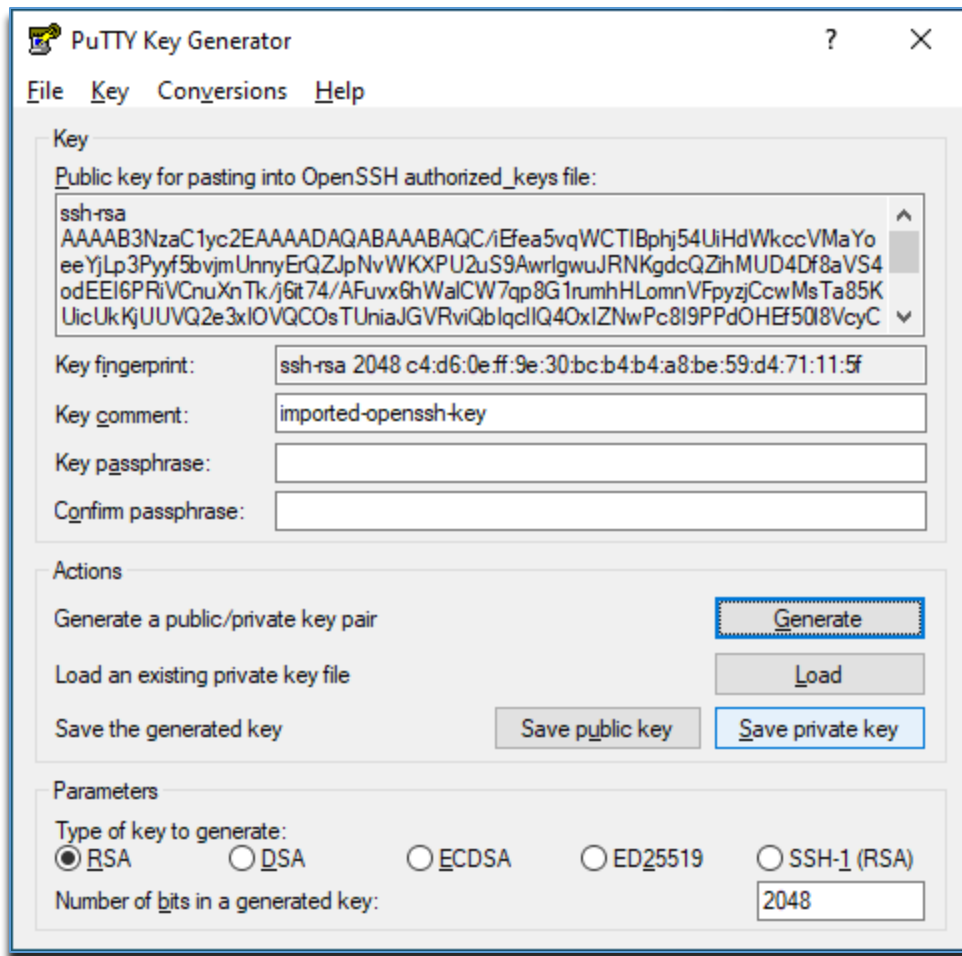
13. Click on View Instances – Allow some time to boot your instance. Once all done, proceed to next to remotely access your newly created instance. You will need PuTTY to do that. Install PuTTY to your local machine from where you intend to access the instance provisioned



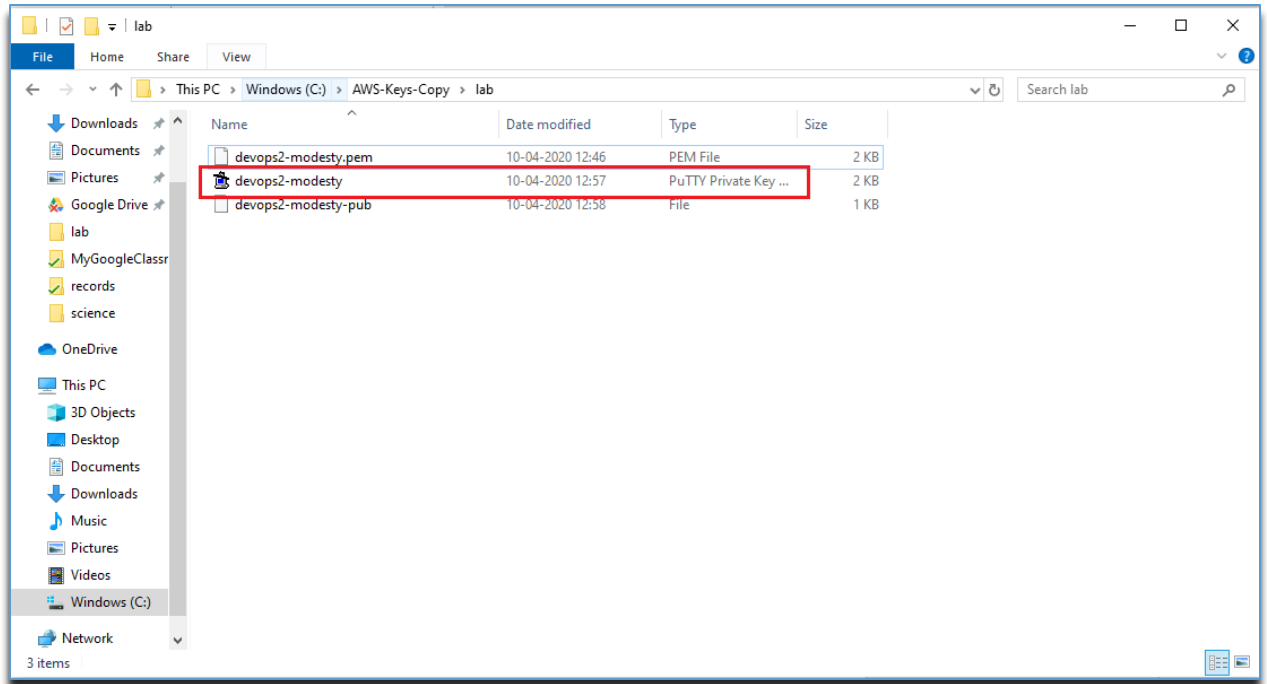
14. To Generate **.ppk file from .pem** file to access the box: Open PuTTYgen from Windows Start / Search menu. Click on **Conversions > Import key** menu



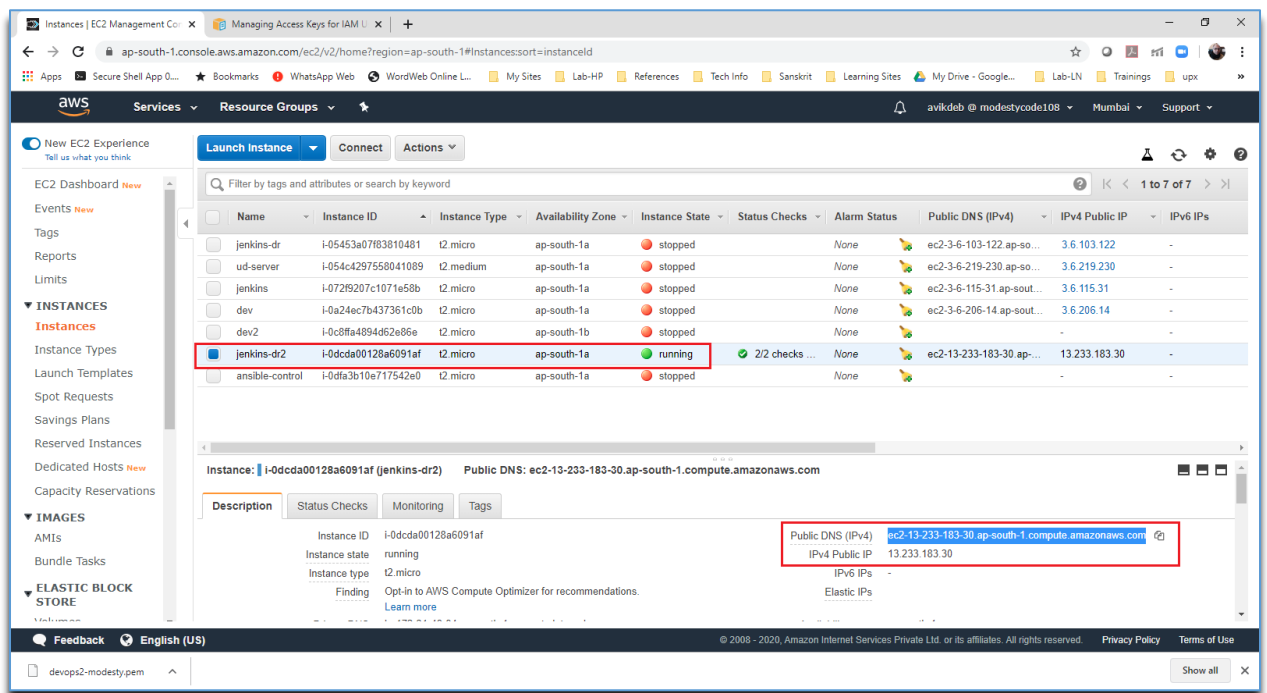
15. Click on Save private key > Give a good meaningful name (suggested to keep the original name as in .pem file) > .ppk file is now created. Similarly save the public key as well

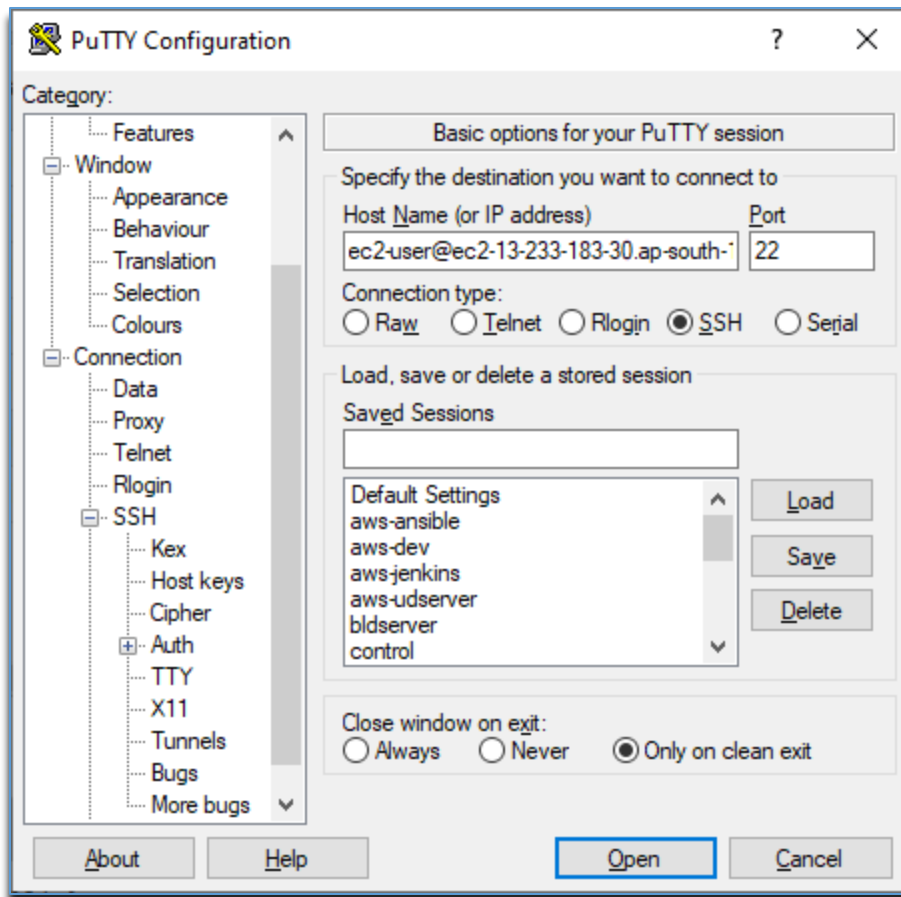


16. Check – you should no have .ppk file created in the desired location in your local machine

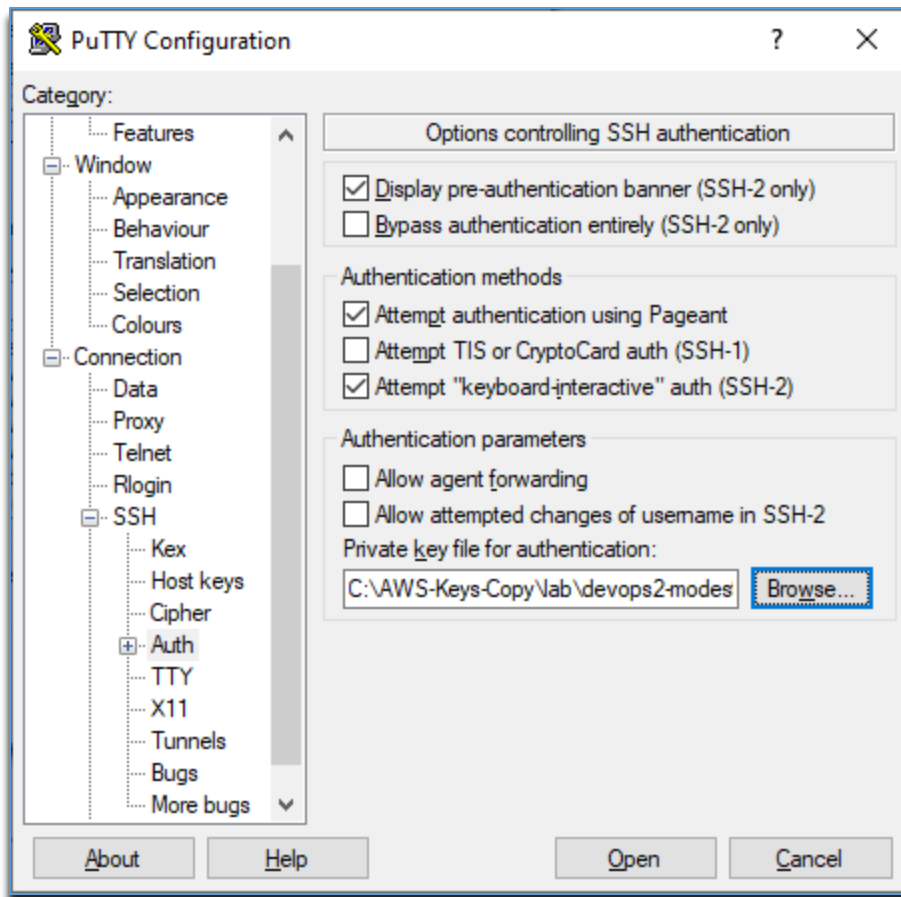


17. Now run PuTTY in your local machine. Provide the connection string as `ec2-user@<public DNS or IP of your AWS EC2 instance>` (This you can get from your AWS management console) . Go to SSH > Auth > Point to the .ppk file location



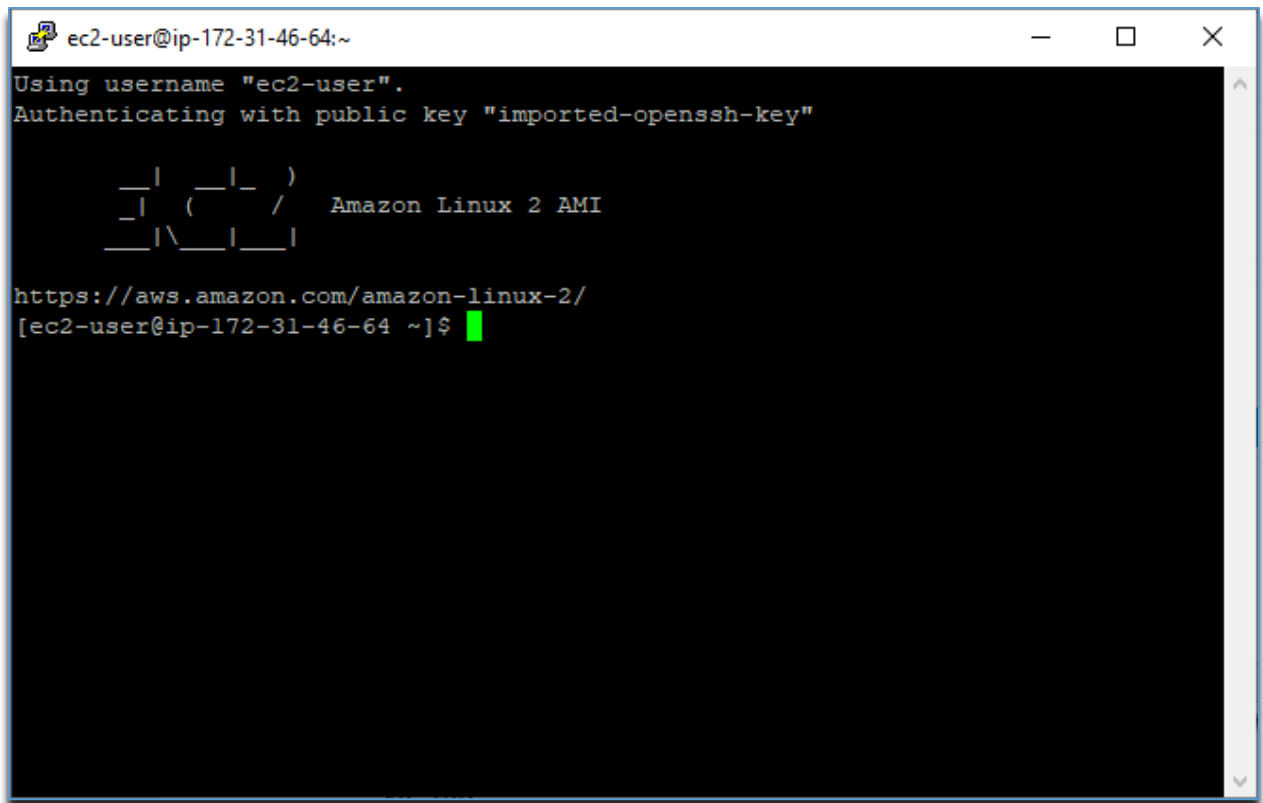


Pointing to the .ppk file generated for ec2-user:



Click **Open**

18. Accept the certificate sent from the remote EC2 box. You should be in to the box now.



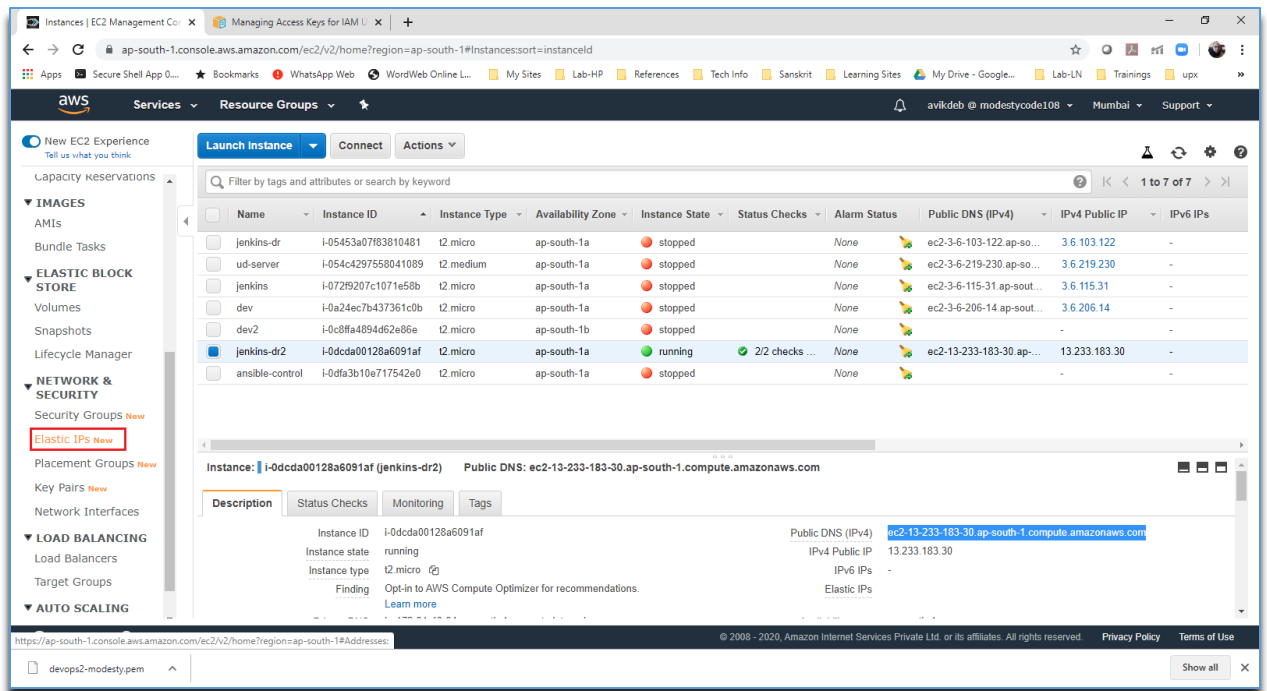
A terminal window titled "ec2-user@ip-172-31-46-64:~" with standard window controls. The terminal output shows the SSH login process: "Using username 'ec2-user'." followed by "Authenticating with public key 'imported-openssh-key'". Below this is the Amazon Linux 2 logo, which consists of a stylized 'A' made of horizontal bars, followed by the text "Amazon Linux 2 AMI". Then, the URL "https://aws.amazon.com/amazon-linux-2/" is displayed. Finally, the prompt "[ec2-user@ip-172-31-46-64 ~]\$" is shown with a green cursor.

```
ec2-user@ip-172-31-46-64:~
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"

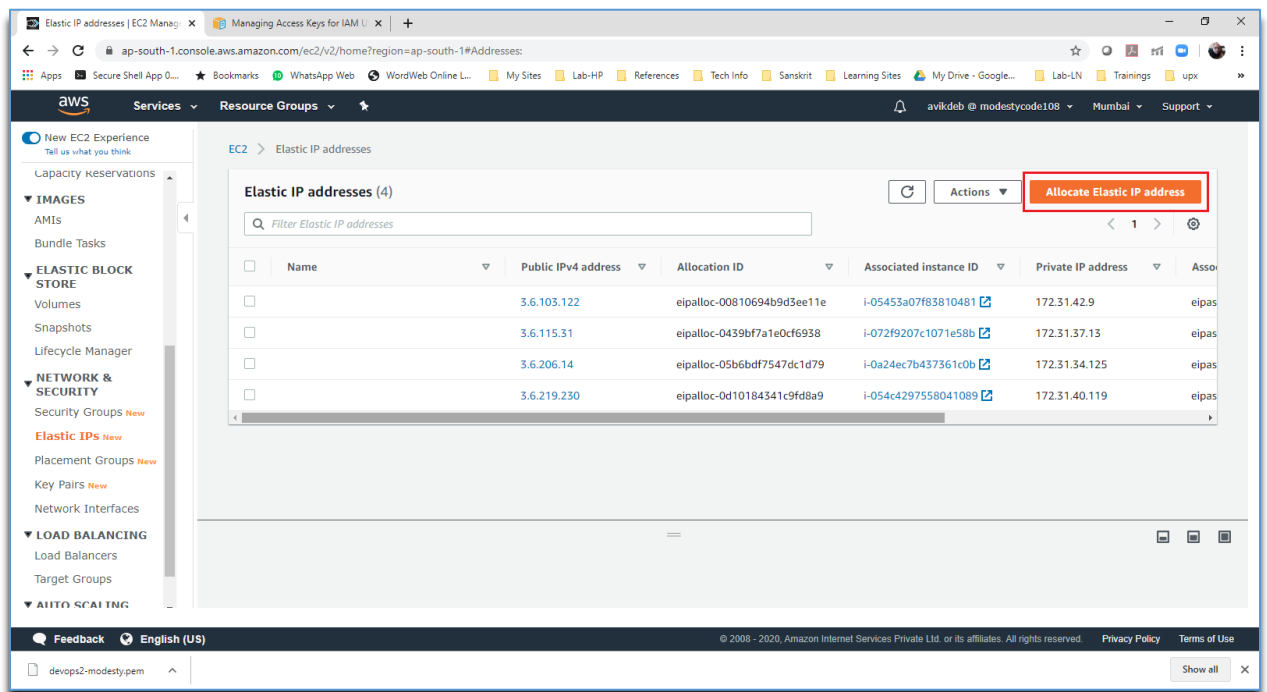
  _ | _ | _ )
  _ | ( _ /   Amazon Linux 2 AMI
  _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-46-64 ~]$
```

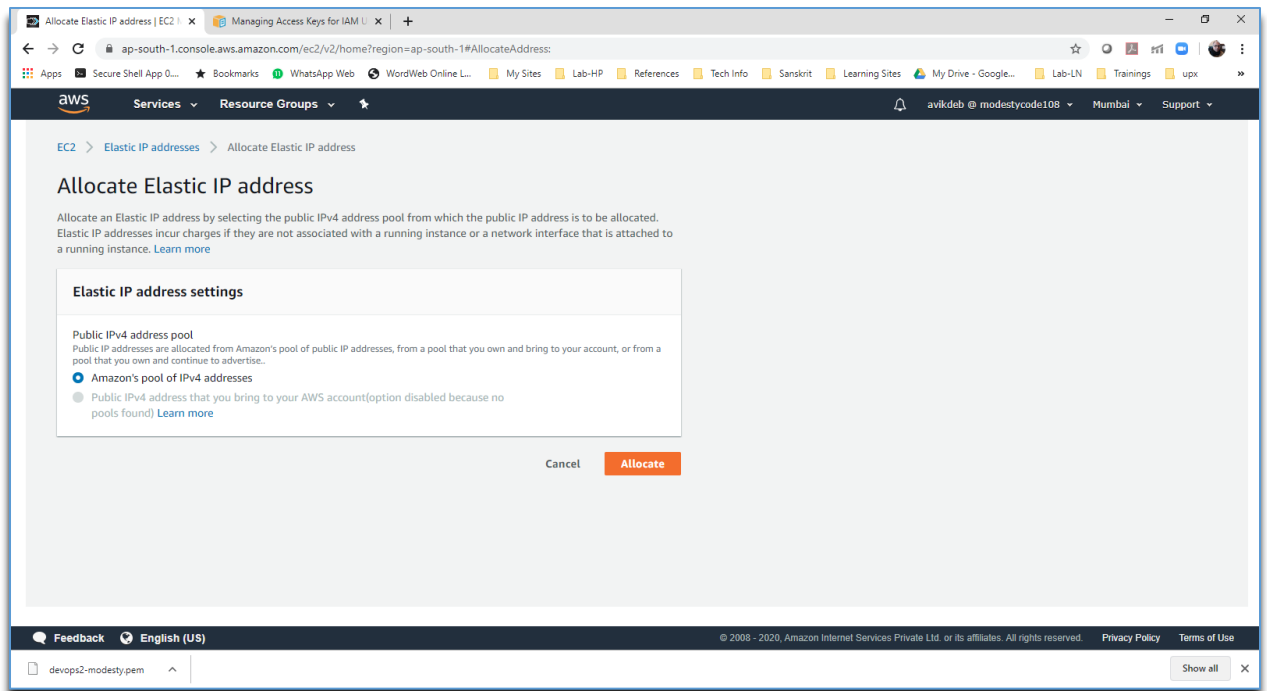
19. Run: `sudo yum update`
20. To assign elastic IP: Let the box be running. In the management console go to the **Network & Security** section in the left side panel > **Elastic IP**



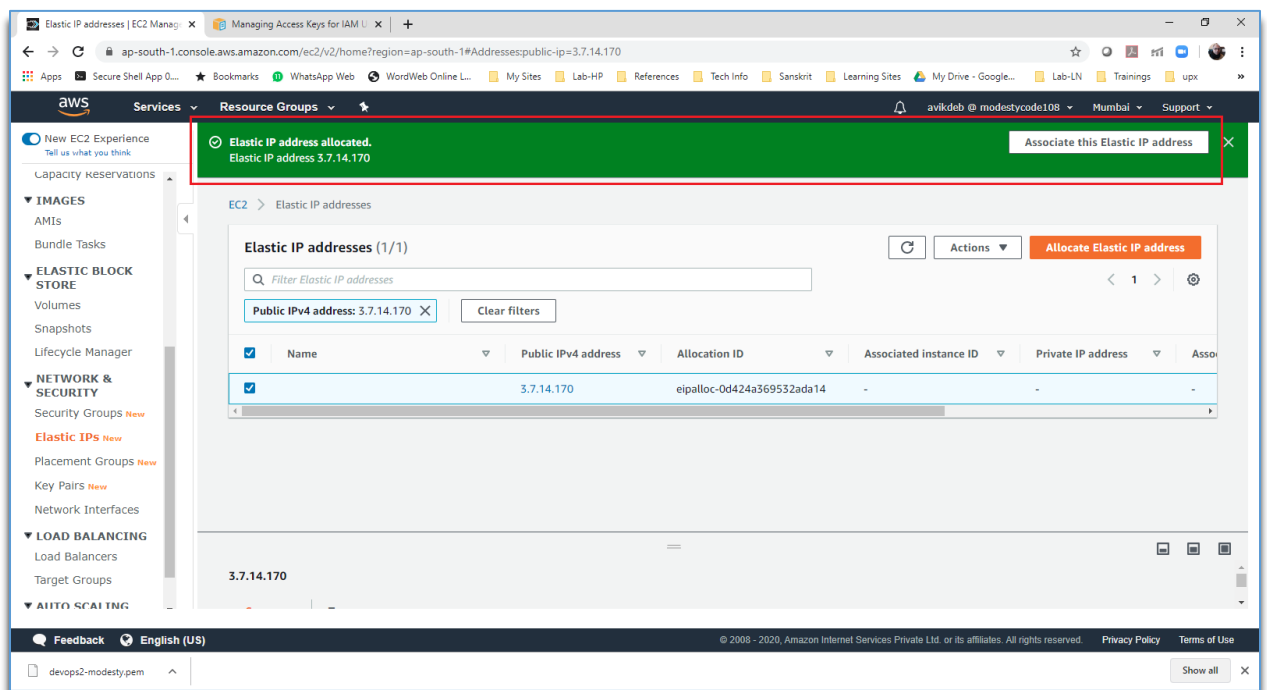
21. Click on **Allocate Elastic IP address** button at the top



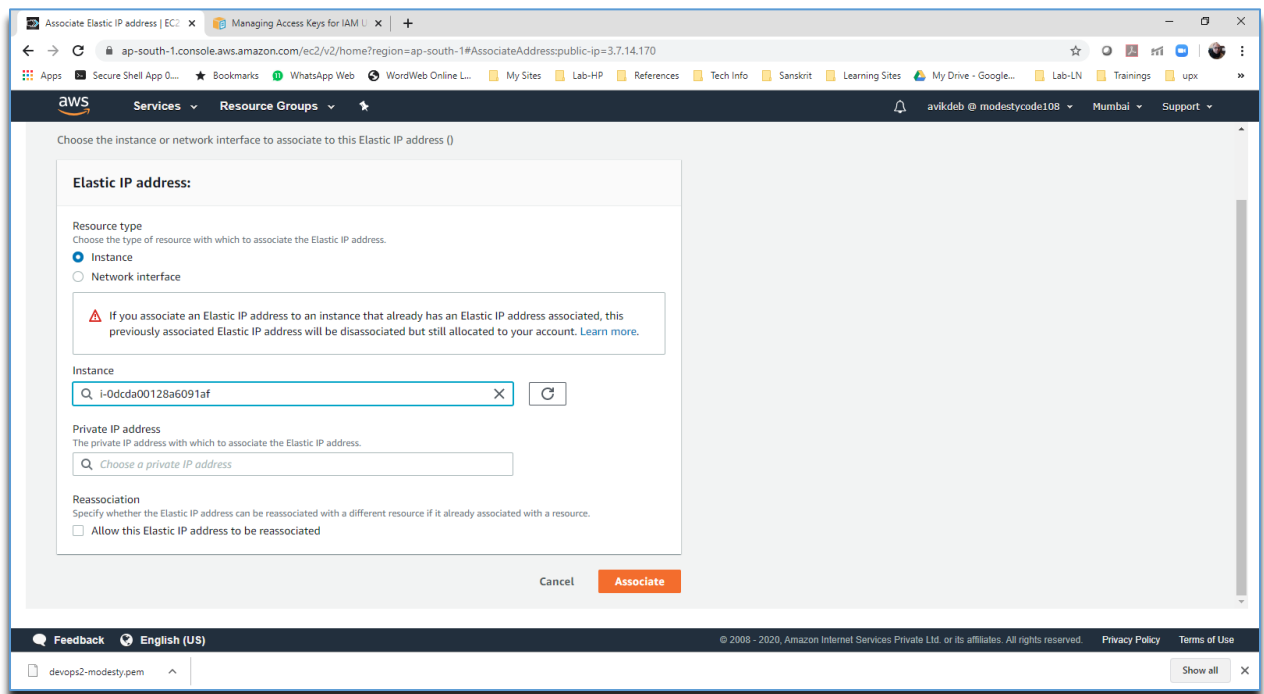
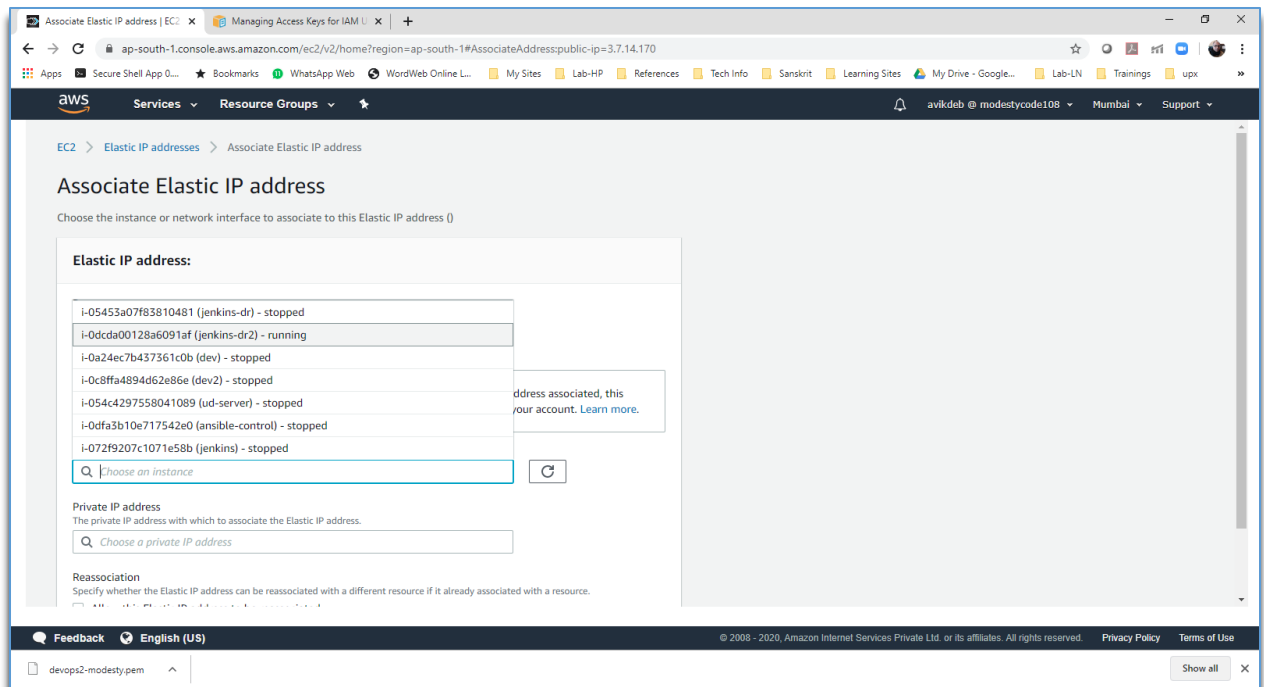
22. Use Amazon's pool of IPv4 addresses. Click on **Allocate** button.



23. Once the IP is allocated, we need to associate this with our EC2 instance. Click on the **Associate this Elastic IP address** button at the top green zone

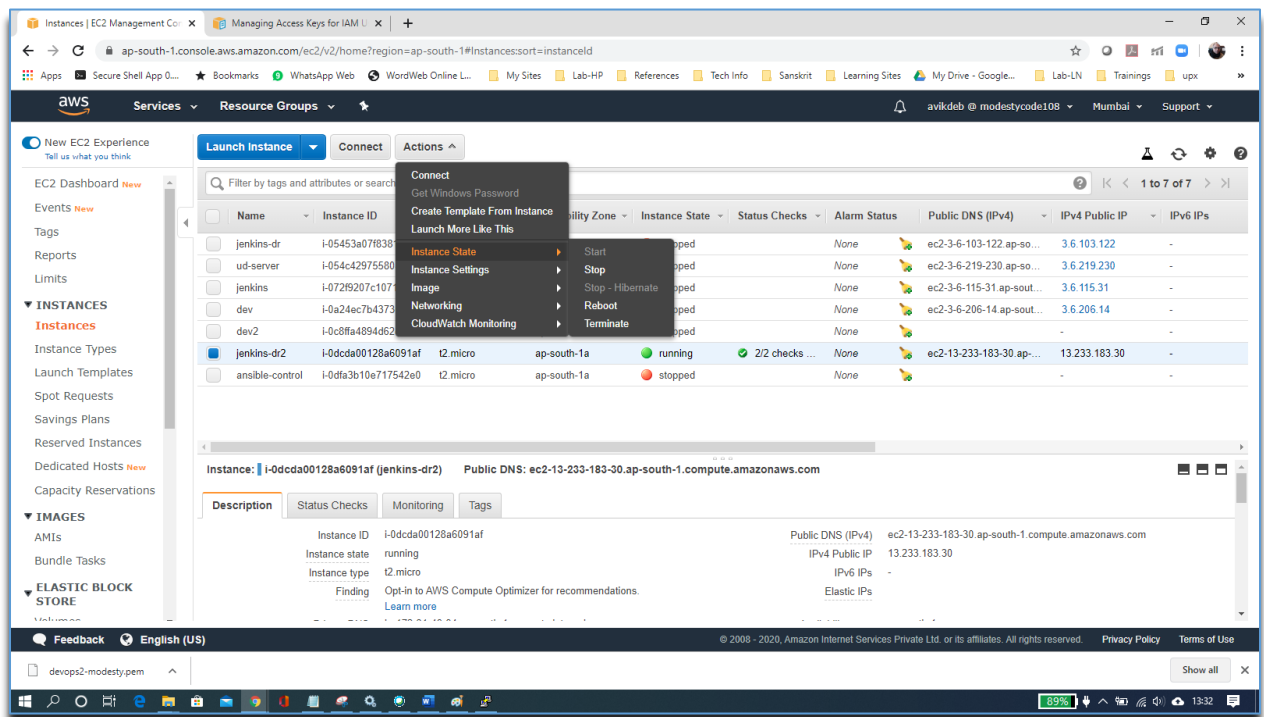


24. Select your instance from the list of instances and click on **Associate** button at the bottom right

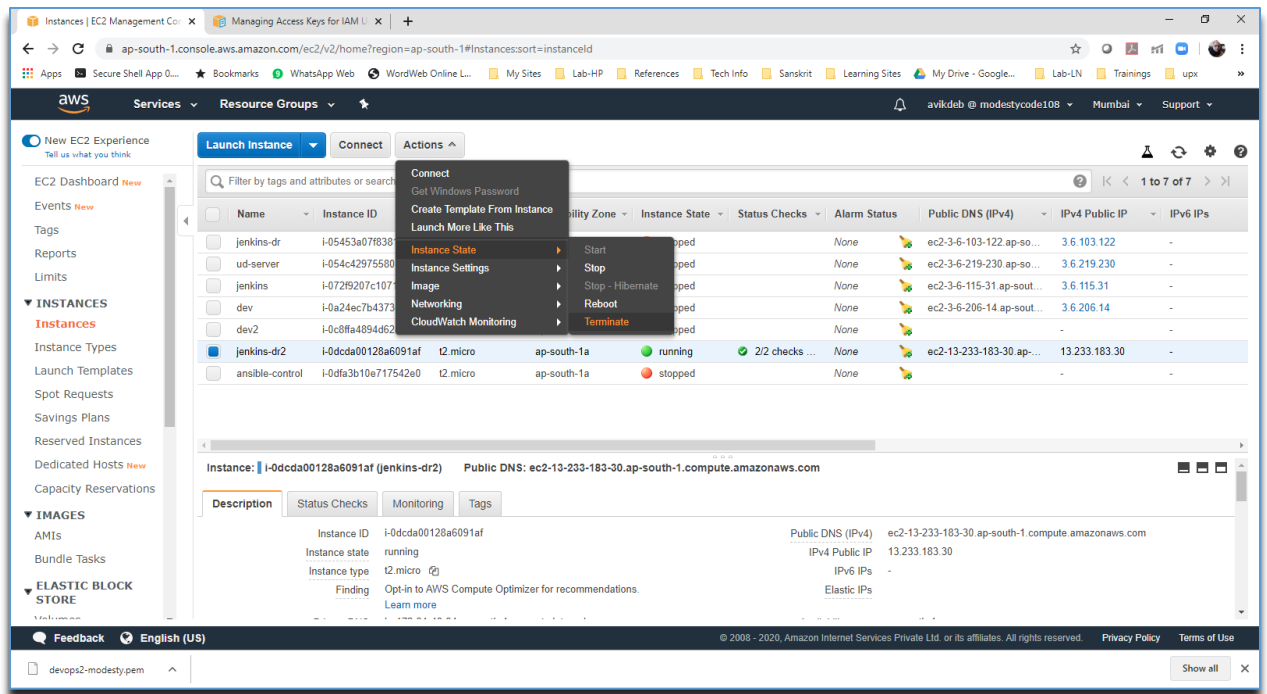


25. Once done, you will have a static IP address associated with your instance. This will not change. Note that Elastic IP is not free and will incur a nominal charge. So, once you are done with the instance do not forget to release the IP address.
26. Your lab is now set up. Spin up as many instances as you want following the same process. You may want to re-use the key pair once you have generated for the first time, in that case during launching a new instance, just select the **Use existing key-pair** option instead of **generate new key-pair**. You should be able to use same .ppk file for all your newly created instances.

27. **To Start / stop your EC2 instance:** In EC2 Instance Dashboard > select your instance > Instance State > Stop / Start (as the case may be)



28. **To Terminate your EC2 instance:** same as Stop / Start but select Terminate instead. This will destroy the box. Release Elastic IP address, if any



Basic Software

- Below is the list of softwares that we will be installing as part of basic installation requirements:

SL	Software Description	Usage
1	Git	Need to do any check-in / check-out from our version control system
3	Nano	User friendly editor for Linux
4	Zip and Unzip utilities	Usually supplied by default. You may have to install this in some rare cases

- To install Git** use command: `sudo install git`
- Once installation is complete, check with command: `git --version`
- To install Nano** use command: `sudo install nano -y`
- Once installation is complete, check with command: `nano <some filename such as hello>`
- Exit from nano editor with `Ctrl+x`. Check out the menus and options given in the bottom part of nano editor