Cloud Architect, Cloud Network Engineer

Compute, Networking

**Lab Steps**

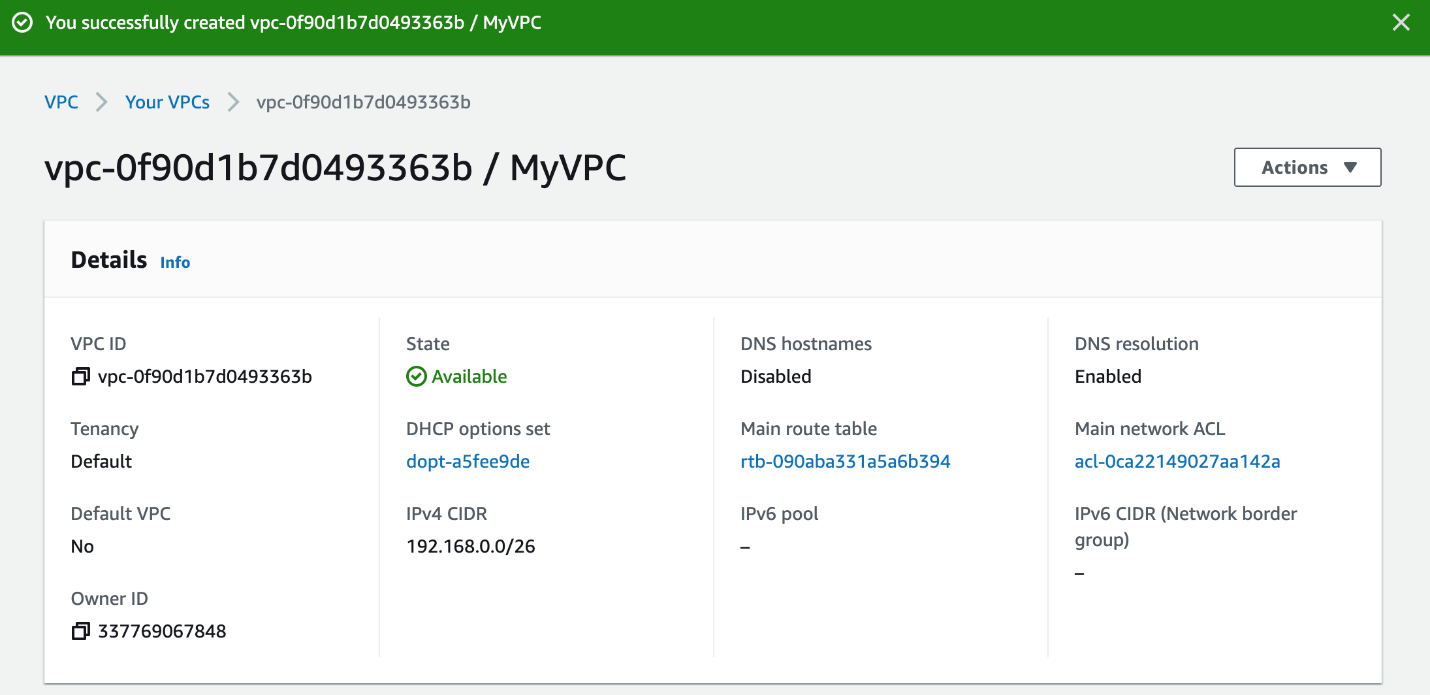
Task 1: Sign in to AWS Management Console

1. Click on the **** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,

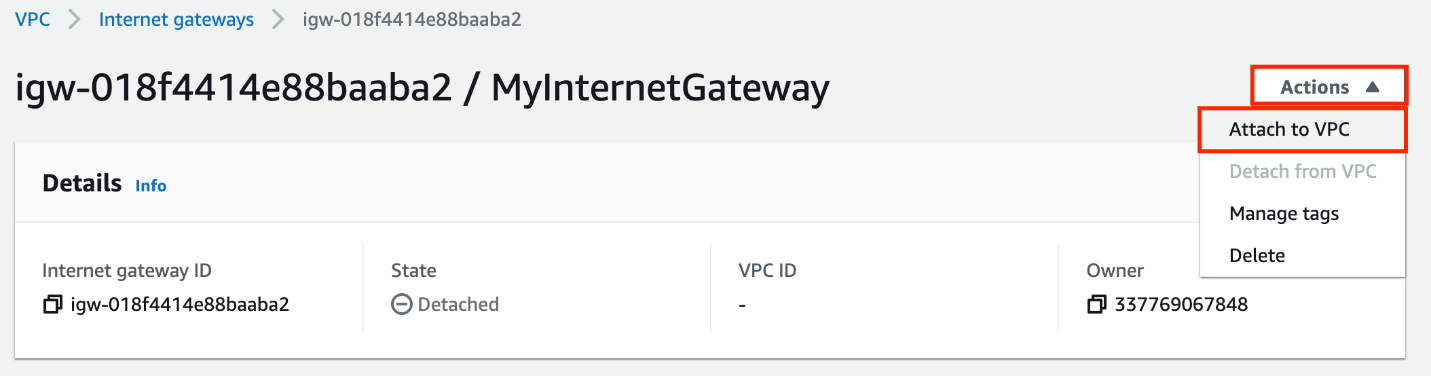
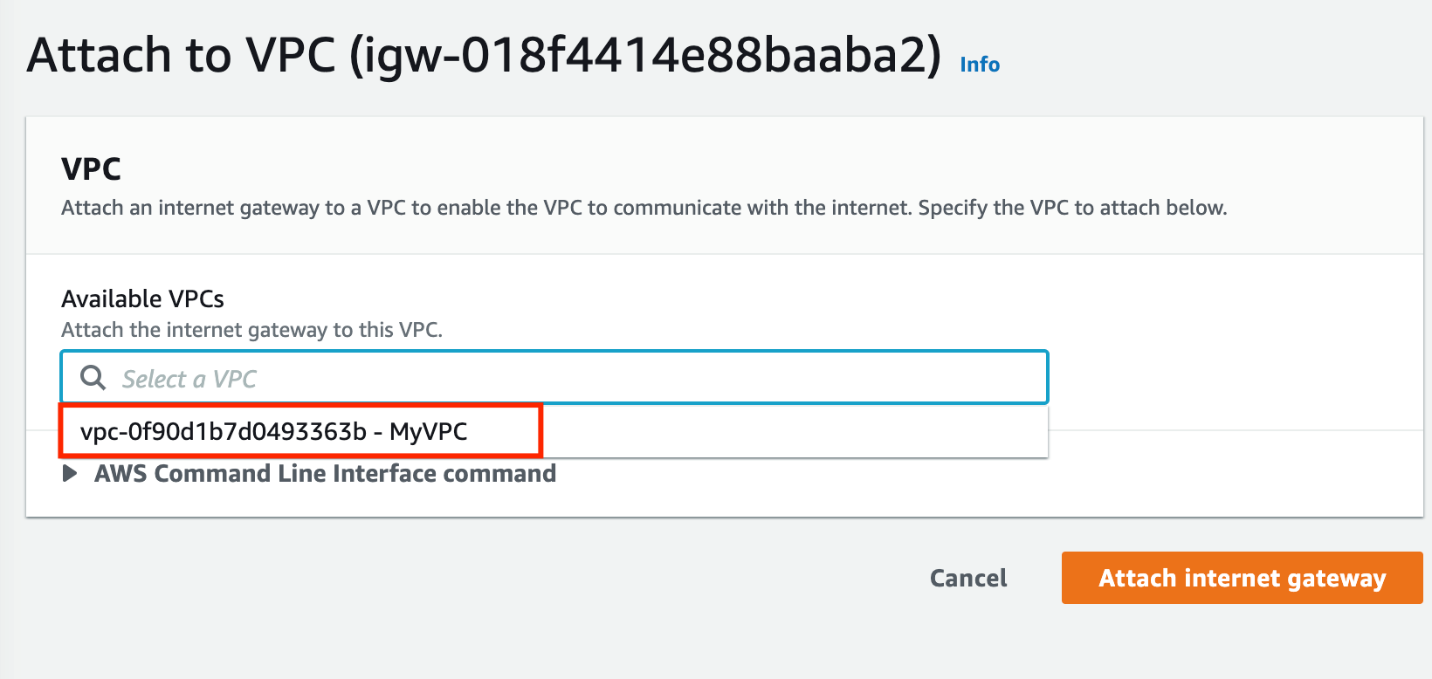
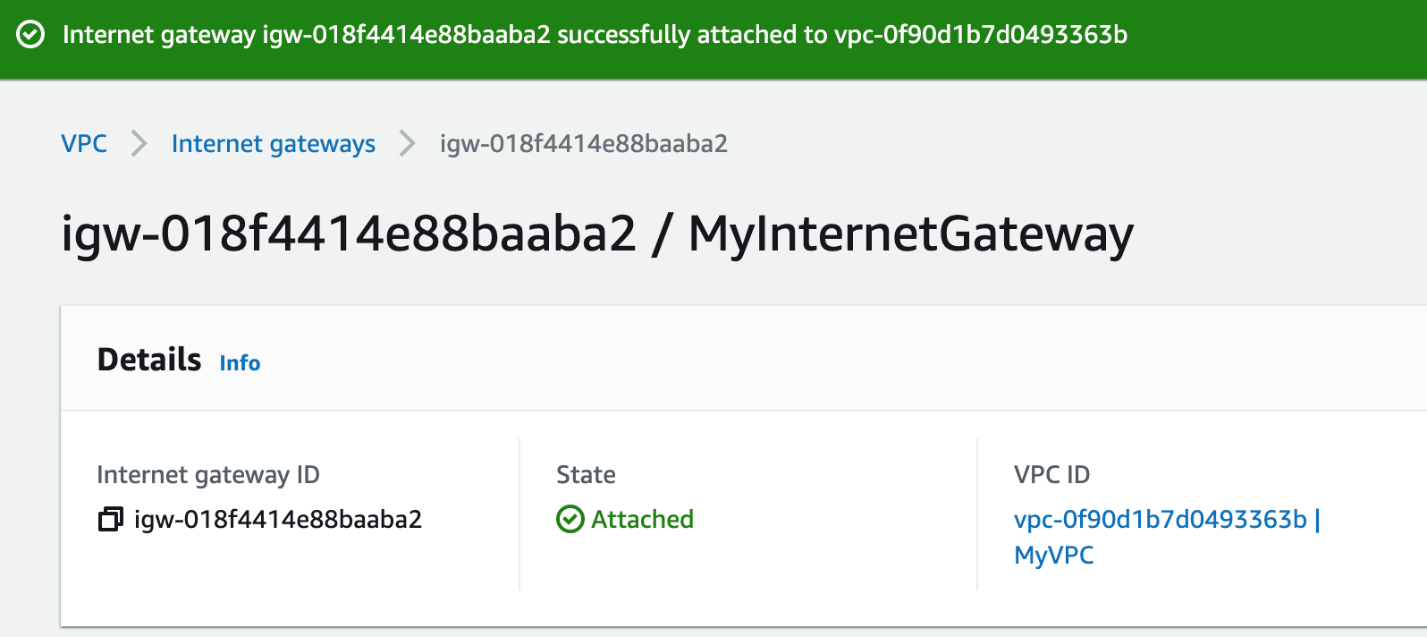
* Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
* Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button

     3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**

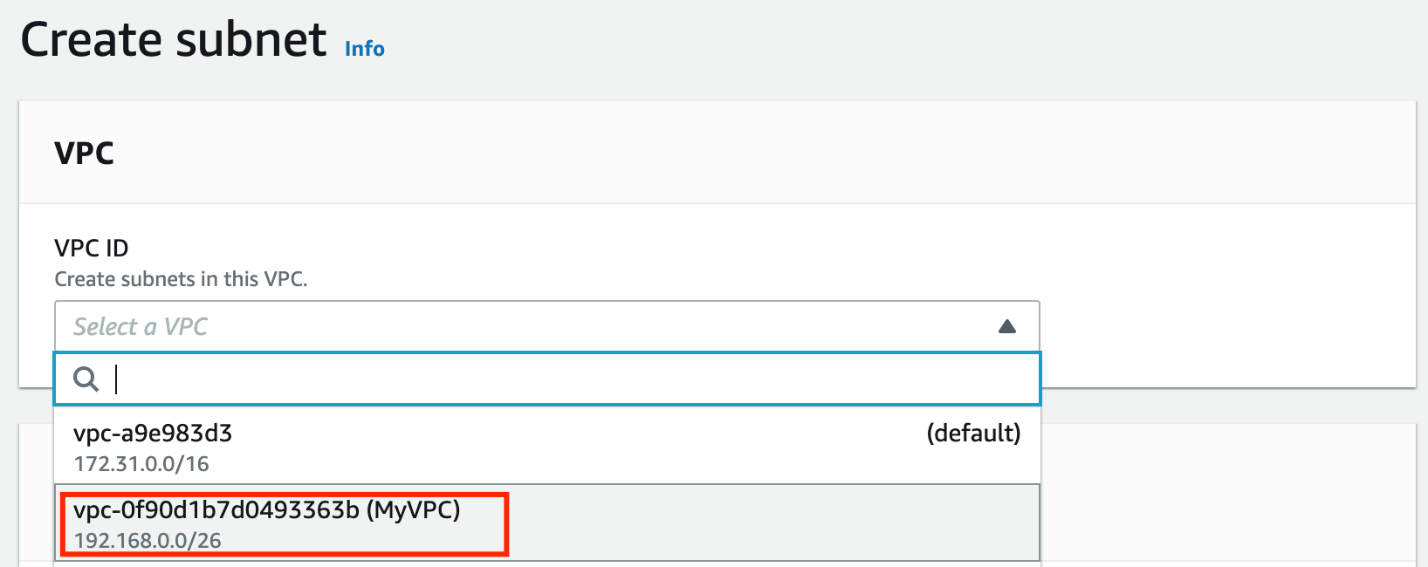
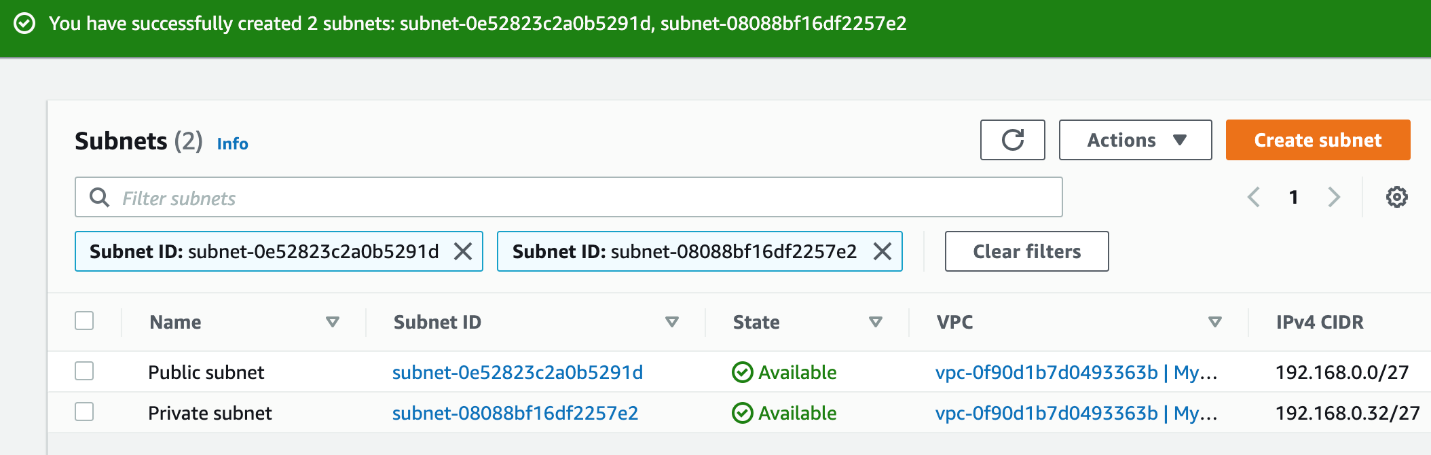
Task 2: Create a VPC

1. Make sure you are in the**N.Virginia**Region.
2. Navigate to **VPC** by clicking on the  menu at the top, then click on **VPC** in the **Network and Content Delivery** section.
3. To create a **VPC** click on  the present in the **VIRTUAL PRIVATE CLOUD** section on the left sidebar.
4. Create a new VPC by clicking on the  .
   * Select **VPC only.**
   * Name tag - optional:Enter ***MyVPC***
   * IPv4 CIDR block: Enter ***192.168.0.0/26***
   * IPv6 CIDR block: No IPv6 CIDR block
   * Tenancy: **Default**
   * Click on the  button to create the **MyVPC**.
5. **VPC is now created.**  
   

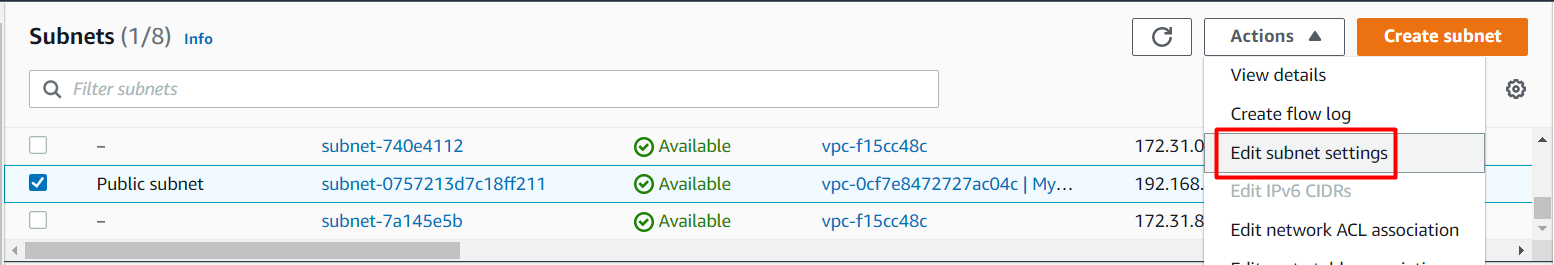
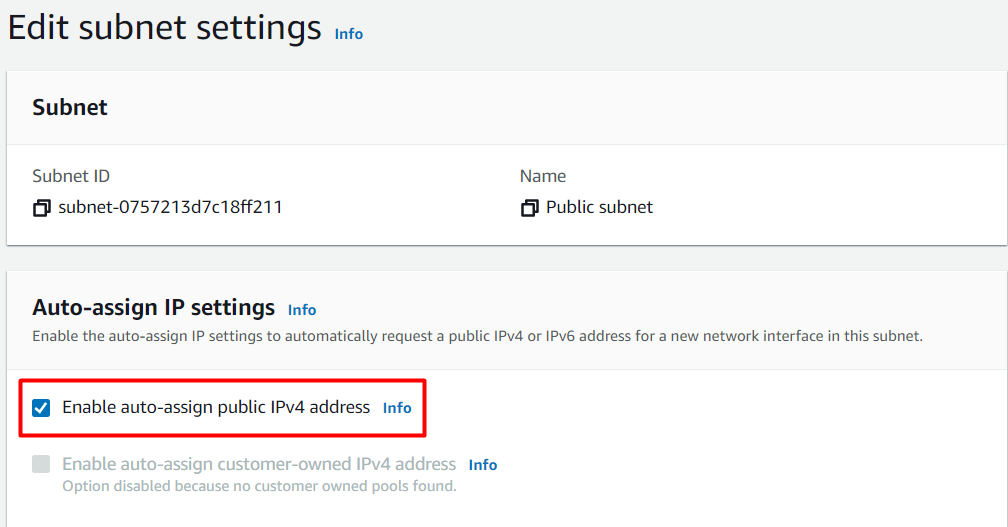
Task 3: Create and attach an Internet Gateway with custom VPC

1. By default, instances that are launched in a VPC cannot communicate with the Internet. To enable Internet access, an Internet gateway needed to be attached to the VPC.
2. Click on **Internet Gateways**from the left menu and click .
   * **Name Tag :** Enter ***MyInternetGateway***
   * Click on .
3. Select the Internet gateway you created from the list.
   * Click on **Actions**.
   * Select the **Attach to VPC**.  
     
4. Available VPCs: Select the **MyVPC**  
   
5. And click on the **Attach internet gateway** button.
6. The Internet gateway is now attached with MyVPC.  
   

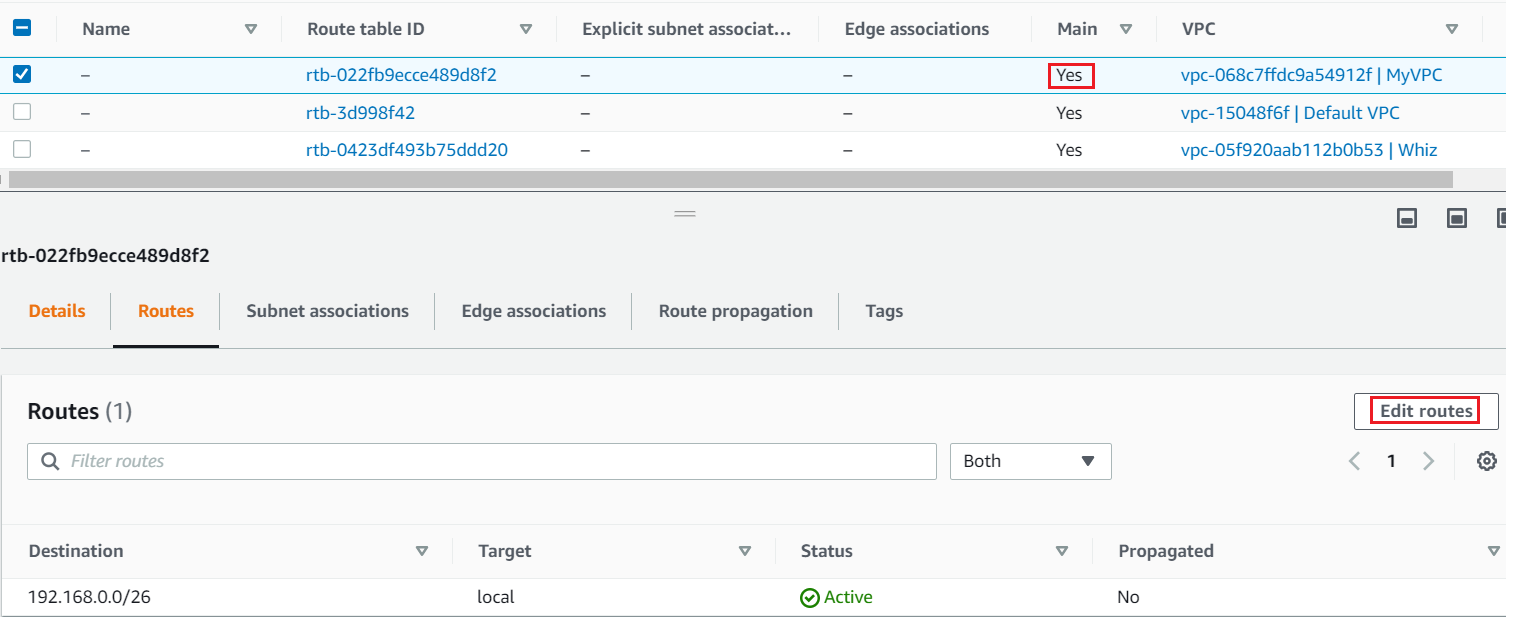
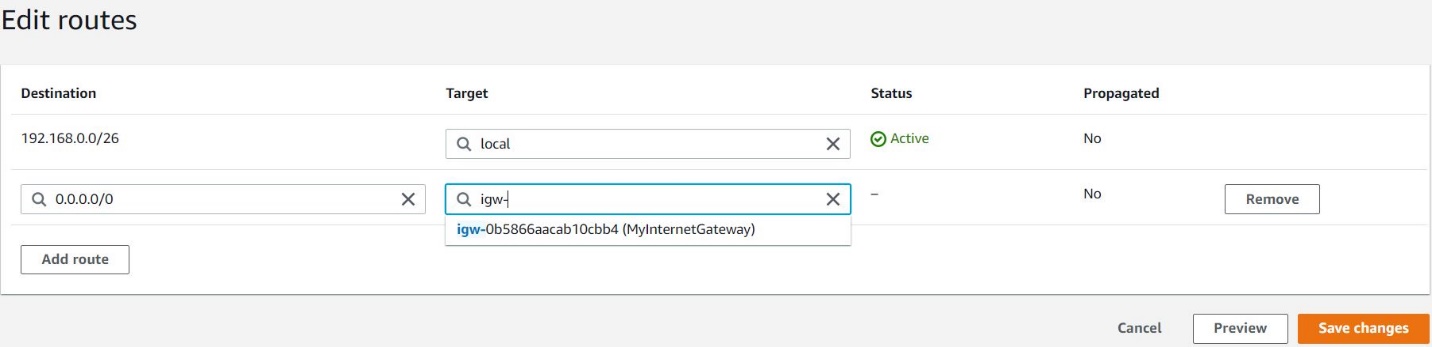
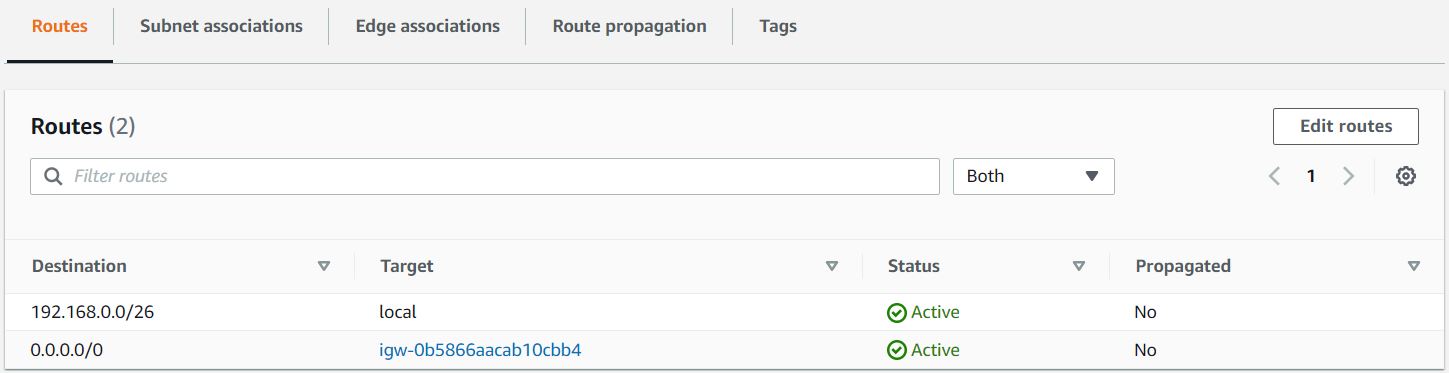
Task 4: Create a Public and Private Subnet

1. To create a **subnet** click on the present in the **VIRTUAL PRIVATE CLOUD** section on the left sidebar.
2. Click on the 
3. In the VPC ID, select the **MyVPC**.  
   
4. Create the first subnet, you will use this subnet to launch public instances, this subnet will be associated with the main route table of the VPC:
   * Select the **MyVPC**from the drop-down.
   * Subnet name: Enter ***Public subnet***
   * Availability Zone: Select **US East (N. Virginia) / us-east-1a**
   * IPV4 CIDR block: Enter ***192.168.0.1/27***
   * Click on the **Create subnet** button to create the subnet.
5. Create another subnet, click on the **Create subnet** button.
6. The second will be called Private subnet, you will use this subnet to launch private instances, this subnet will be associated with a custom route table of the same VPC:
   * Select the **MyVPC**from the drop-down.
   * Subnet name: Enter ***Private subnet***
   * Availability Zone: Select **US East (N. Virginia) / us-east-1b**
   * IPV4 CIDR block: Enter ***192.168.0.32/27***
7. Finally, click on the   to create **subnets**.
8. Both the subnets are now created.  
   

Task 5: Configure the Public subnet to enable auto-assign public IPv4 address

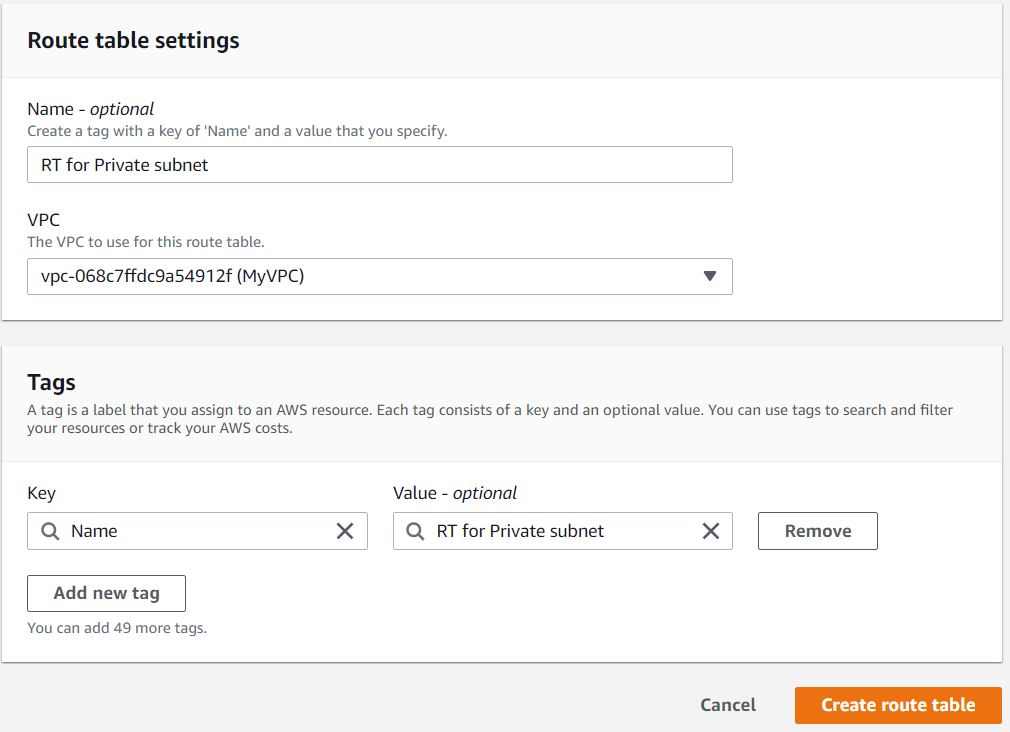
1. To modify the auto-assign IP settings for the Public subnet, do the following:
   * Select the **Public subnet**
   * Click on the **Actions** button
   * Choose **Edit subnet settings** from the options.  
     
2. Check the option**Enable auto-assign public IPv4 address** under **Auto-assign IP settings.**  
   
3. Click on **Save**button and modification is done now.  
   

Task 6: Add an entry to the Internet (0.0.0.0/0) in the Main Route table.

1. By default, custom VPC's main Route Table will have access to VPC's CIDR range only. For this lab, we need internet access.
2. Let's add the route of the internet gateway as destination and 0.0.0.0/0 as Target.
3. To add the route entry of the internet gateway in the main route table, do the following:
   * Select **Route tables**from the left panel.
   * Select the main route table.
   * Click on the **Routes** button in the menu bar.
   * Choose the **Edit routes** option.  
     
4. On the Edit routes page, Click on the **Add route** button.
5. Add the **Destination** as ***0.0.0.0/0*** and Select the **Internet gateway** present.  
   
6. Click on the **Save changes** button, to confirm the entry of the Internet gateway in the Main Route Table.
7. Routes are now edited.
8. Close the pane, and Check the Routes of this Main Route Table. Entry to the Internet i.e. 0.0.0.0/0 via Internet gateway is present in the Routes.  
   

Task 7: Create a Route Table for the Private subnet

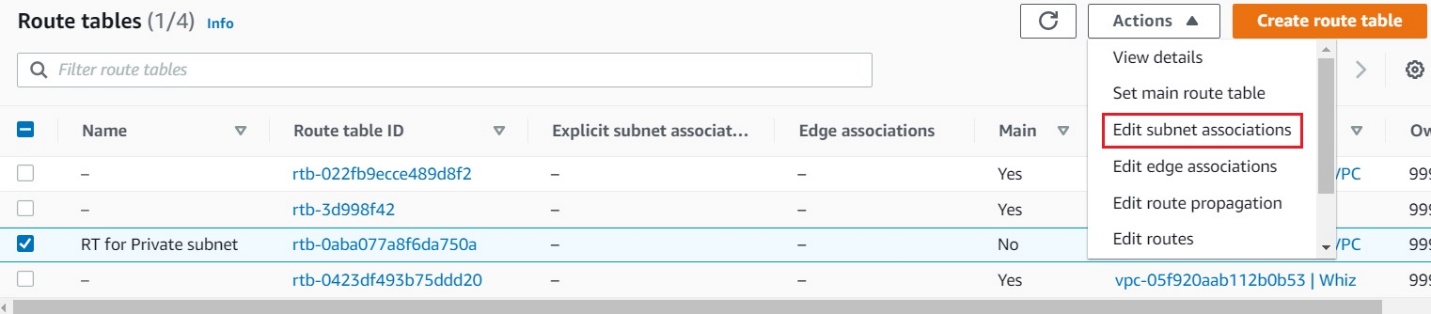
1. Navigate to the **Route tables** section present in the **VIRTUAL PRIVATE CLOUD** section on the left sidebar.
2. To create a Route table, click on the **Create route table** button.
3. Fill in the below details:
   * Name tag: Enter ***RT for Private subnet***
   * VPC:Select**MyVPC**

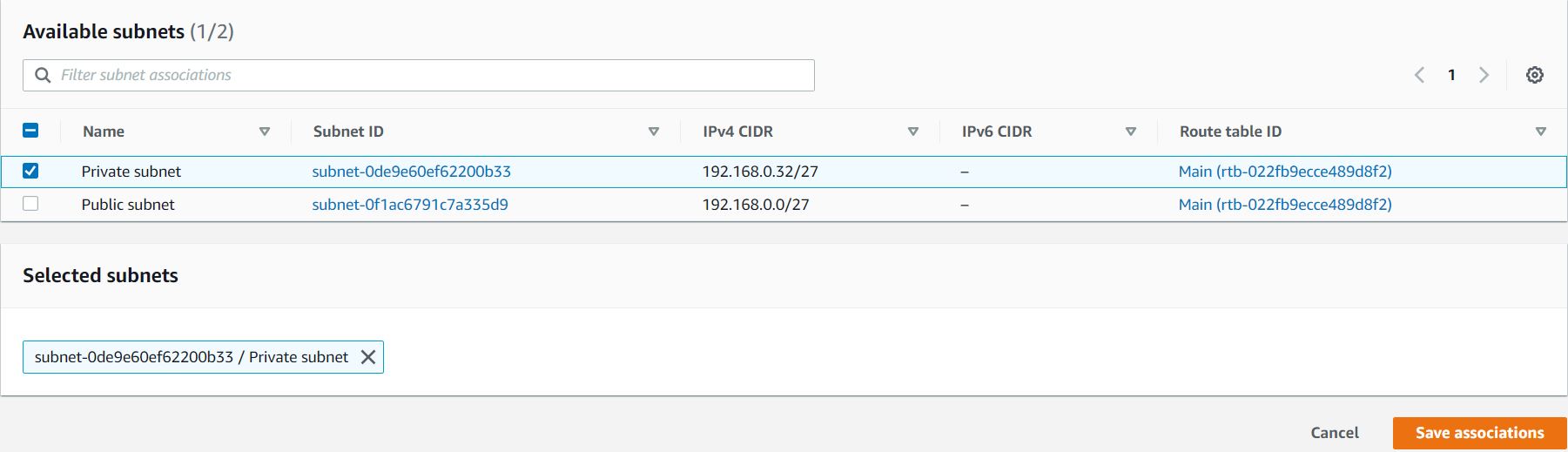
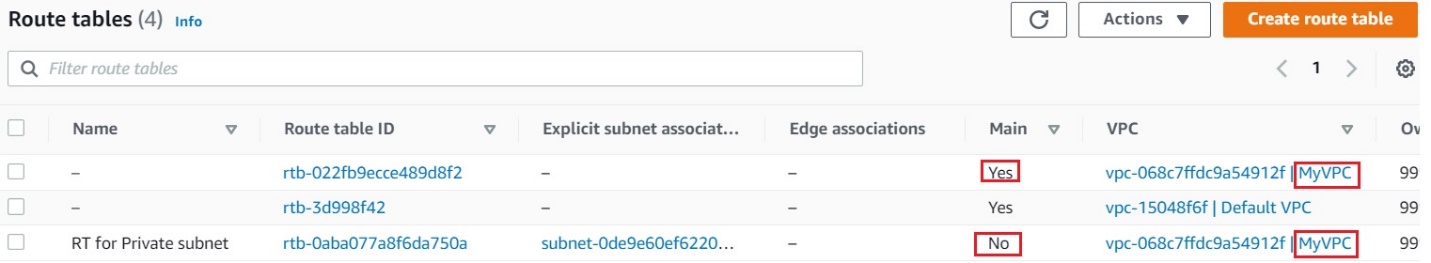


1. Click on the **Create route table** button, present below.

Task 8: Associate Private subnet with the custom Route table i.e. RT for Private subnet

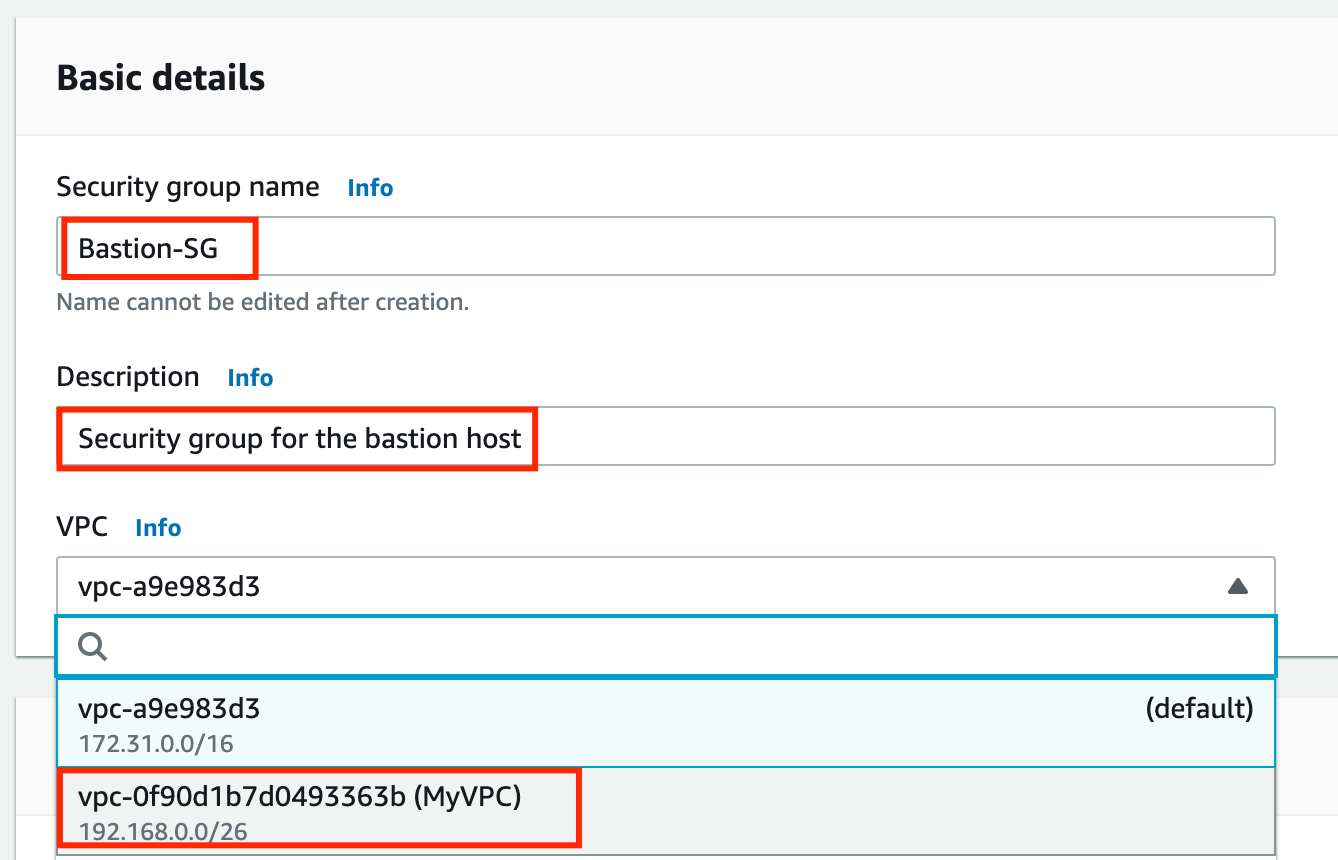
1. To associate Private subnet with a custom route table, do the following:
   * **Select** the custom route table i.e. **RT for Private subnet**
   * Click on the **Actions** button
   * Choose **Edit subnet associations** from the options.



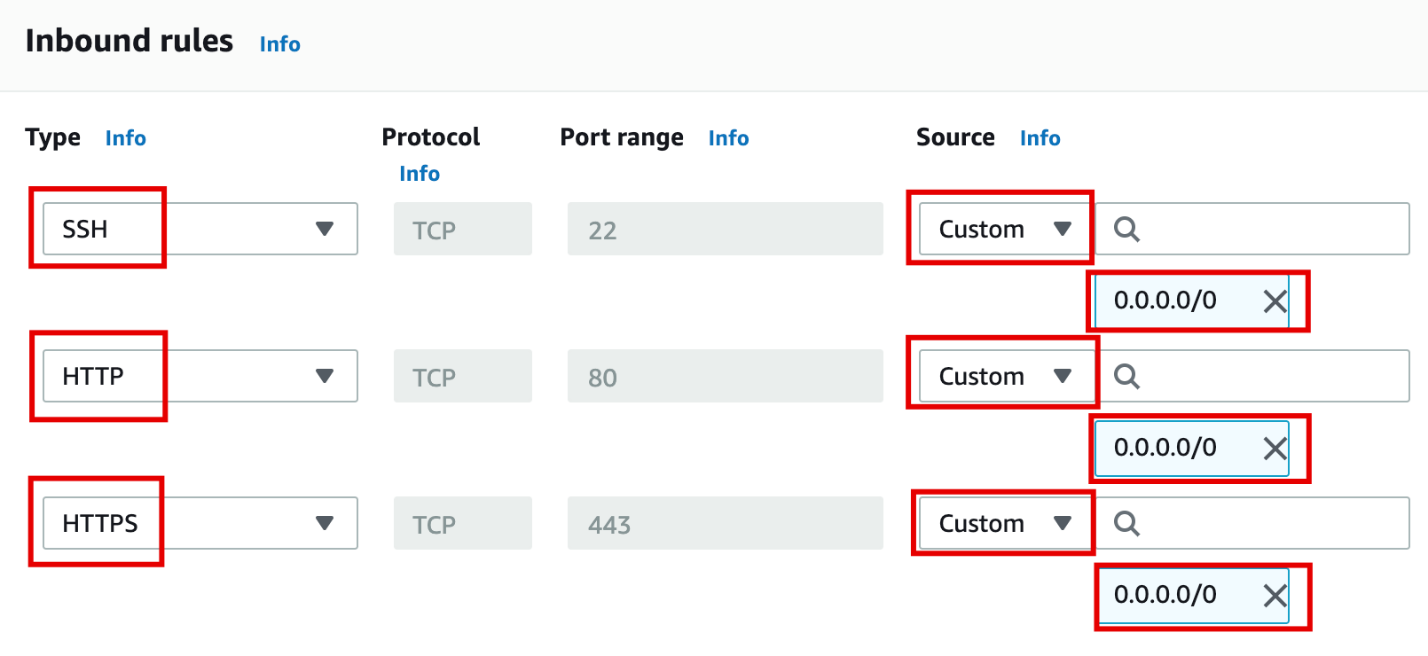
1. Select the subnet with a name ending as **Private subnet** from the **Subnet ID** column. And, Click on the **Save associations** button.  
   
2. The private subnet is now associated with our custom route table i.e. **RT for Private subnet**.
3. One more thing to notice here is, you can call the Route Table a custom route when it is having a **No** in the **Main** column. And, the other one is the **Main route table** as it is created by default when you create a **custom VPC**.  
   

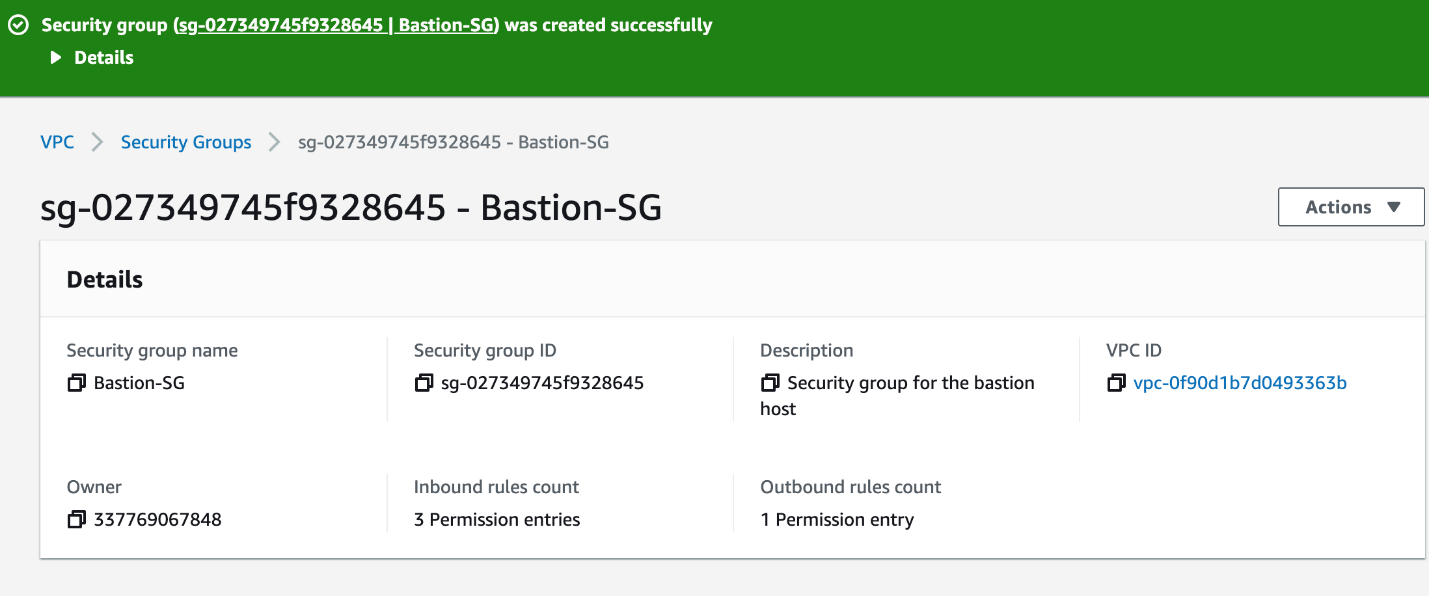
Task 9: Create Security groups

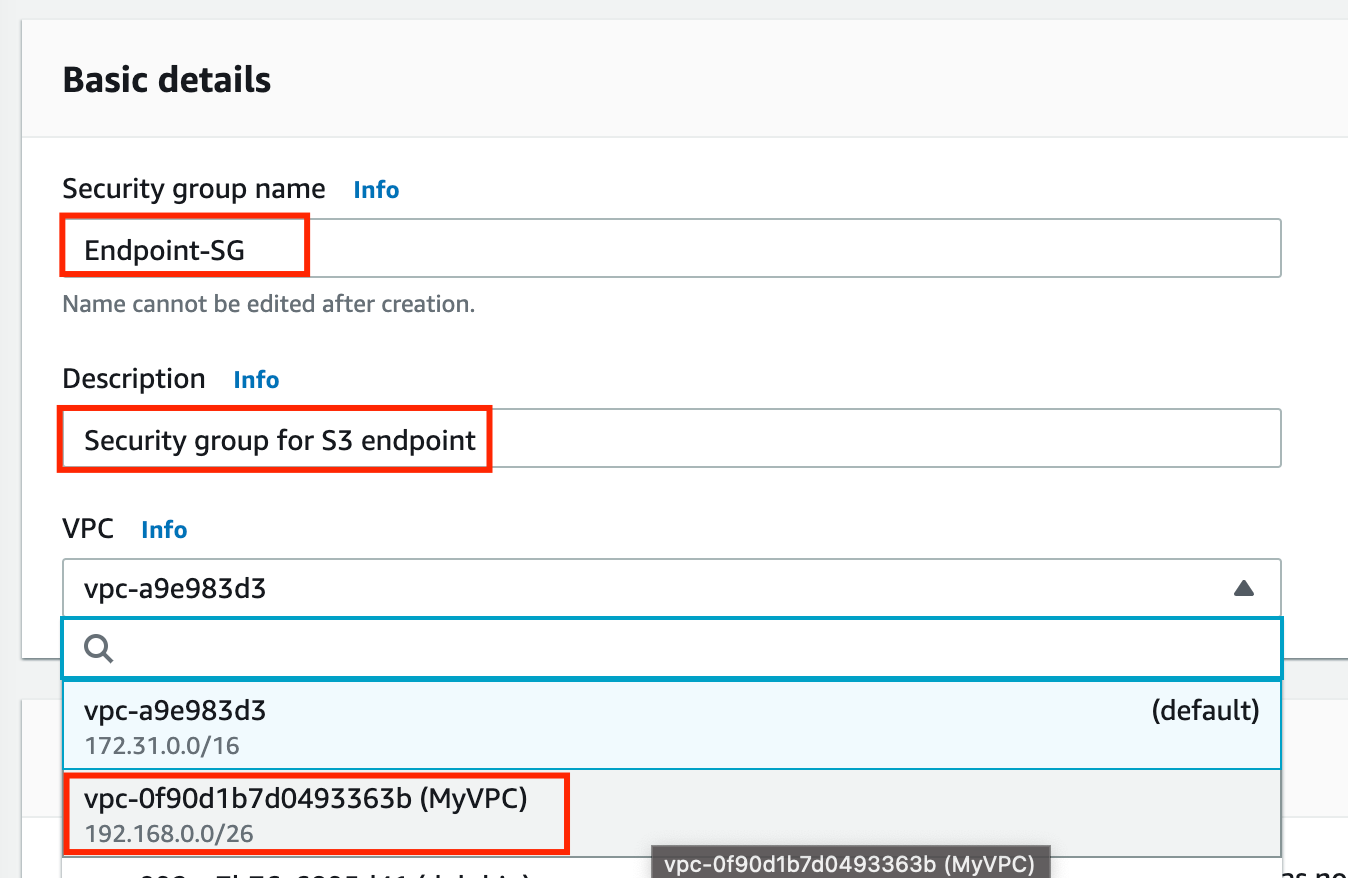
1. In this lab, we will create two security groups, the **first one will be used for Bastion host**and the **second one for private instance having access to VPC Endpoint for S3.**
2. To get started with creating security groups, click on the  , present in the **SECURITY** section in the left sidebar.
3. Click on the 
4. Fill in the below details under **Basic details**:
   * Enter **Security group** name as ***Bastion-SG***
   * Enter **Description** as ***Security group for the bastion host***
   * Select the**MyVPC** under the **VPC** field.

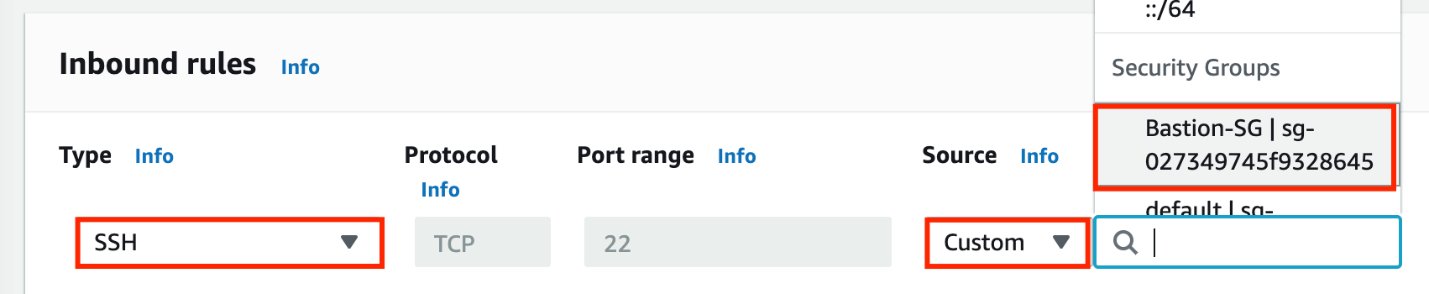
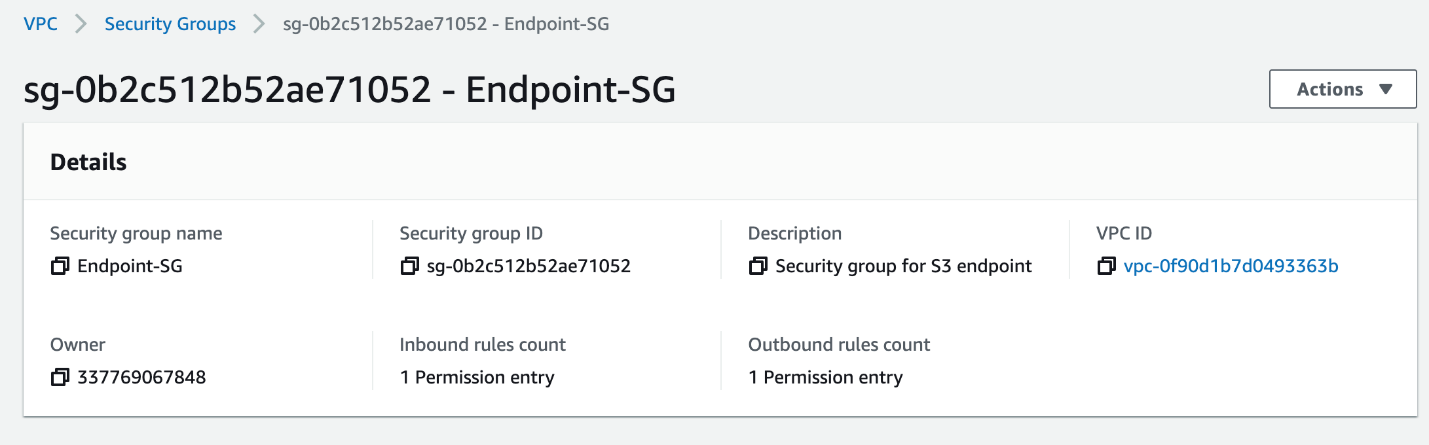


1. By default, no inbound rule will be allowed, and when you check in the outbound rules, there is only one rule present that has a Type will **All traffic** because **Security groups are Stateful** in nature, **when inbound is allowed outbound is also allowed.**
2. To add the Inbound rules for the same, click on the  button.
3. We will add **3 rules** for the Bastion host security group i.e. **SSH, HTTP, and HTTPS.**
   * For the first rule, Select the **Type** as ***SSH***, **Source** as **Custom** and **enter *0.0.0.0/0***
   * For the second rule, click on the  button. Select the **Type** as ***HTTP***, **Source** as **Custom** and **enter *0.0.0.0/0***
   * For the third rule, click on the  button. Select the **Type** as ***HTTPS***, **Source** as **Custom** and **enter *0.0.0.0/0***



1. Finally, click on the   to create.
2. The security group for the Bastion host is now created.  
   
3. Click on the  button, present in the **SECURITY** section in the left sidebar.
4. To create the second security group, click on the 
5. Fill in the below details under **Basic details**:
   * Enter **Security group** name as ***Endpoint-SG***
   * Enter **Description** as ***Security group for S3 endpoint***
   * Select the**MyVPC** under the **VPC** field.



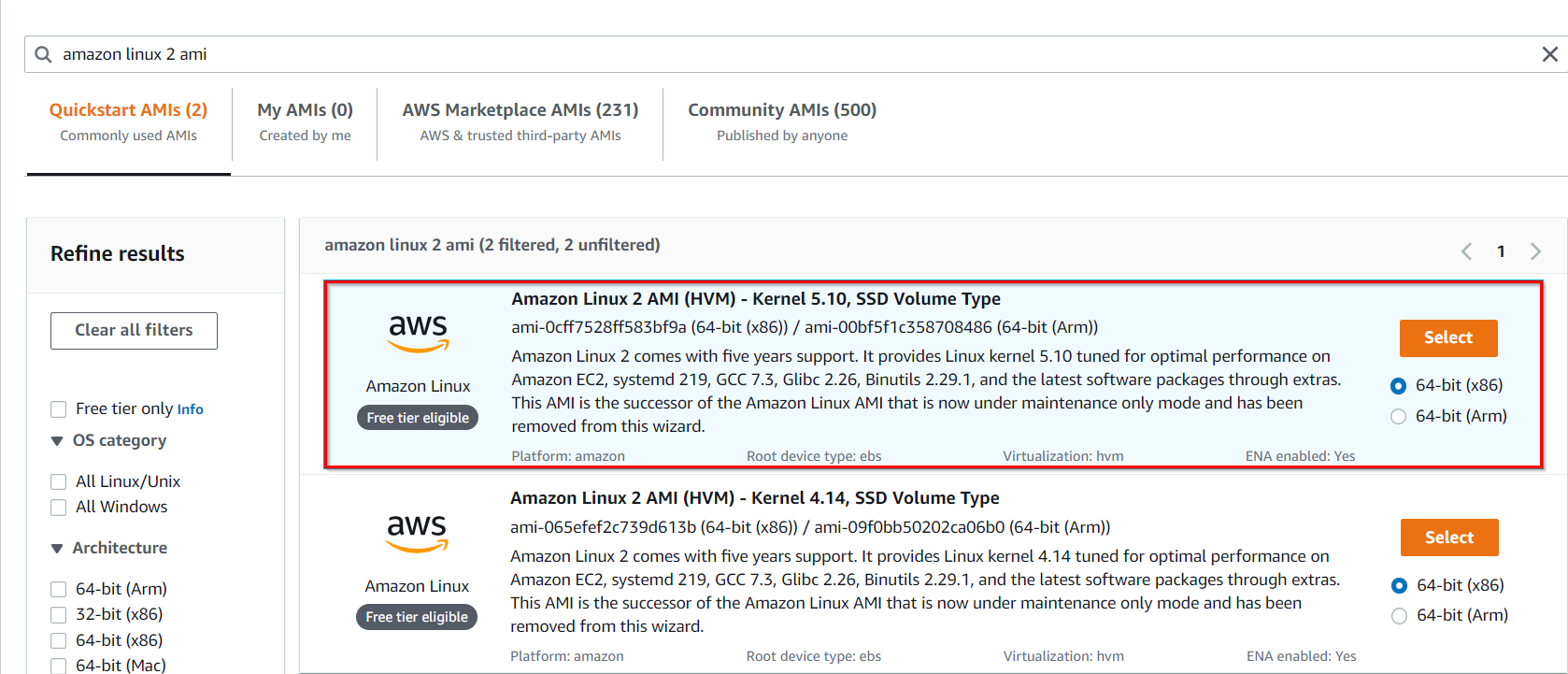
1. To add the Inbound rules for the same, click on the  button.
2. We will add **1 rule** for the S3 endpoint security group i.e. **SSH *only*.**But here our source will be **Bastion host security group**, you will select the**ID of Bastion-SG**security group.
   * For the rule, select the **Type** as ***SSH***, **Source** as **Custom,** and **type *Bastion*,**Bastion hosts security group will be shown, select that Security group.  
     
3. Finally, click on the  to create.
4. The security group for the Bastion host is now created. And, it will be listed there.  
   

Task 10: Create a Bastion host (Publicly accessible EC2 Instance)

  1. Navigate to **EC2** by clicking on the  menu at the top, then click on **EC2** in the **Compute** section.

  2. Navigate to **Instances** on the left panel and click on **Launch instances.**

  3. Name : Enter ***Bastion-host.***

   4. **For Amazon Machine Image (AMI):** Search for **Amazon Linux 2 AMI** in the search box and click on the **Select**button.  
        

     5 . **Note: if there are two AMI's present for Amazon Linux 2 AMI, choose any of them.**

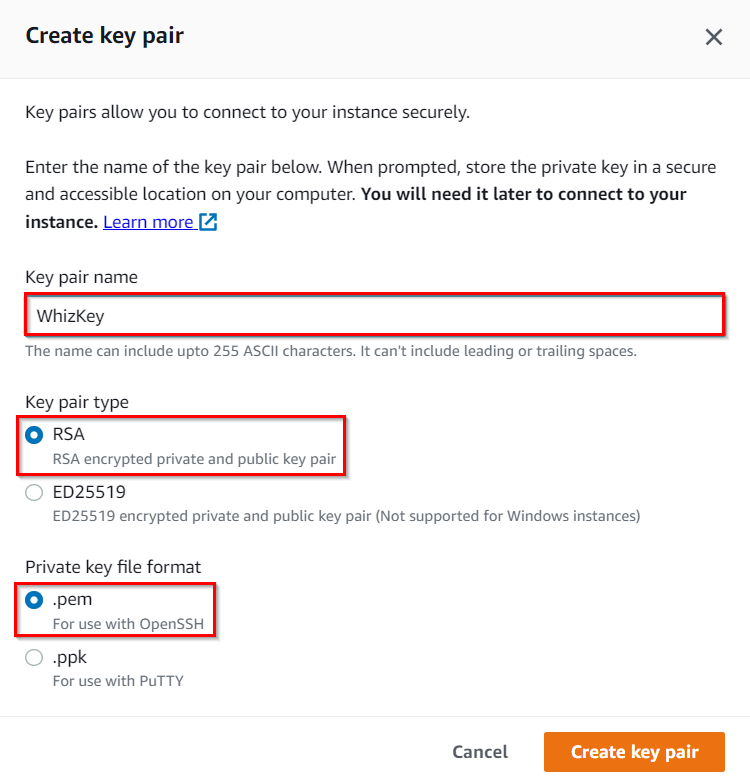
     6. **For Instance Type:** select ***t2.micro***

****

 7.**For Key pair:**Select **Create a new key pair**Button

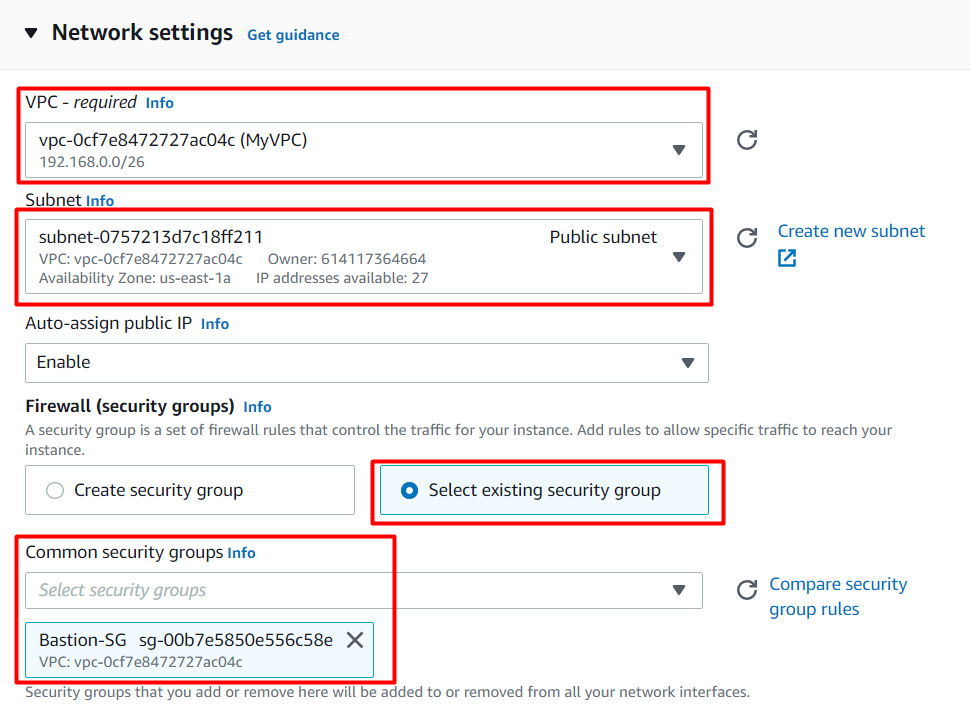
* Key pair name: **WhizKey**
* Key pair type:**RSA**
* Private key file format: **.pem**

1. Select **Create key pair** Button.



     9.  In Network Settings Click on **Edit** Button:

* VPC: Choose **MyVPC**
* Subnet : Choose **Public Subnet**
* Auto-assign public IP: **Enable**
* Select **existing security group**
* Security group name : Choose **Bastion-SG**

**

   10. Keep rest thing default and Click on**Launch Instance b**utton.

   11. Select **View all Instances** to view Instance you created

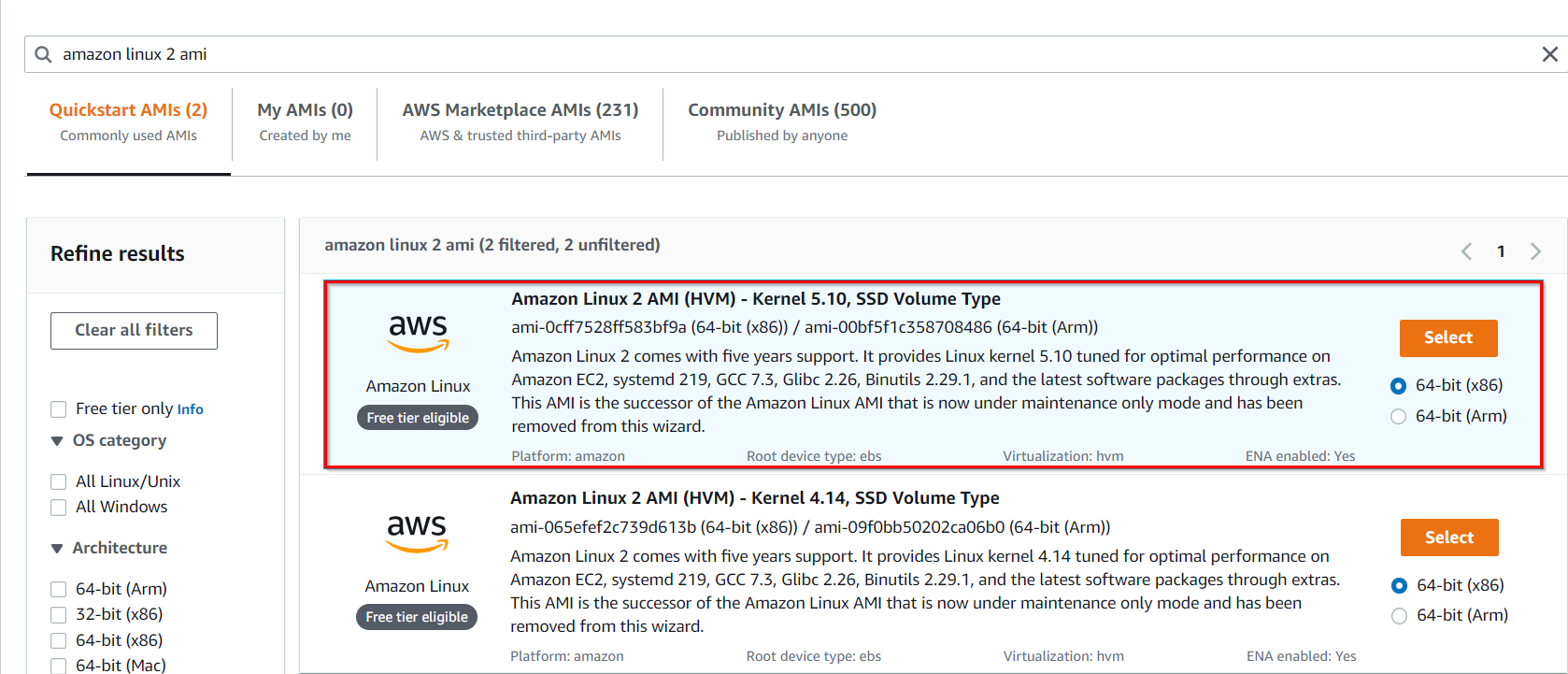
   12. **Launch Status:** Your instance is now launching, Click on the instance ID and wait for complete initialization of the instance till status changes to **Running**.

Task 11: Create an Endpoint instance (Privately accessible EC2 instance)

  1. Navigate to **EC2** by clicking on the  menu at the top, then click on **EC2** in the **Compute** section.

  2. Navigate to **Instances** on the left panel and click on **Launch instances.**

  3. Name : Enter ***Endpoint-instance.***

   4. **For Amazon Machine Image (AMI):** Search for **Amazon Linux 2 AMI** in the search box and click on the **Select**button.  
        

     5 . **Note: if there are two AMI's present for Amazon Linux 2 AMI, choose any of them.**

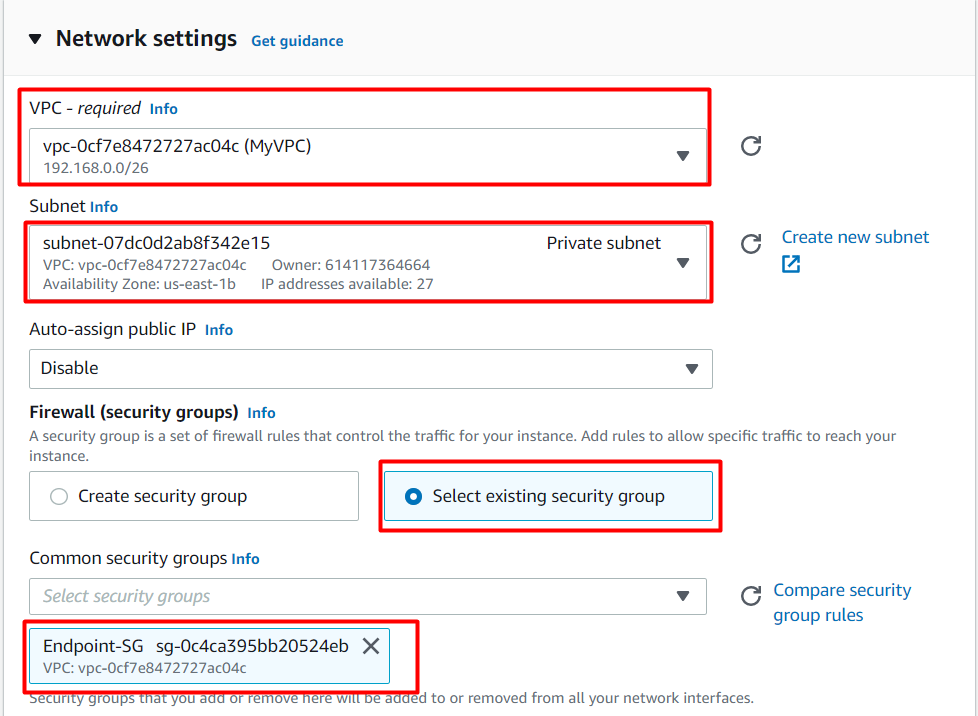
     6. **For Instance Type:** select ***t2.micro***

****

 7.**For Key pair:**Select the key pair made during previous task.

     8.  In Network Settings Click on **Edit** Button:

* VPC: Choose **MyVPC**
* Subnet : **Private Subnet**
* Select **existing security group**
* Security group name : Choose **Endpoint-SG**

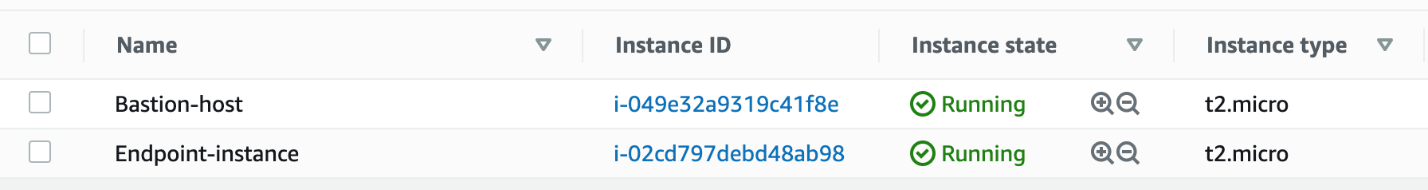
**

   10. Keep rest thing Default and Click on**Launch Instance b**utton.

   11. Select **View all Instances** to view Instance you created

   12. **Launch Status:** Your instance is now launching, Click on the instance ID and wait for complete initialization of the instance till status changes to **Running**.

   13. Navigate to **Instances** and wait for 1-2 minutes (until the **Endpoint-instance's** status changes from **pending** to **running**state)



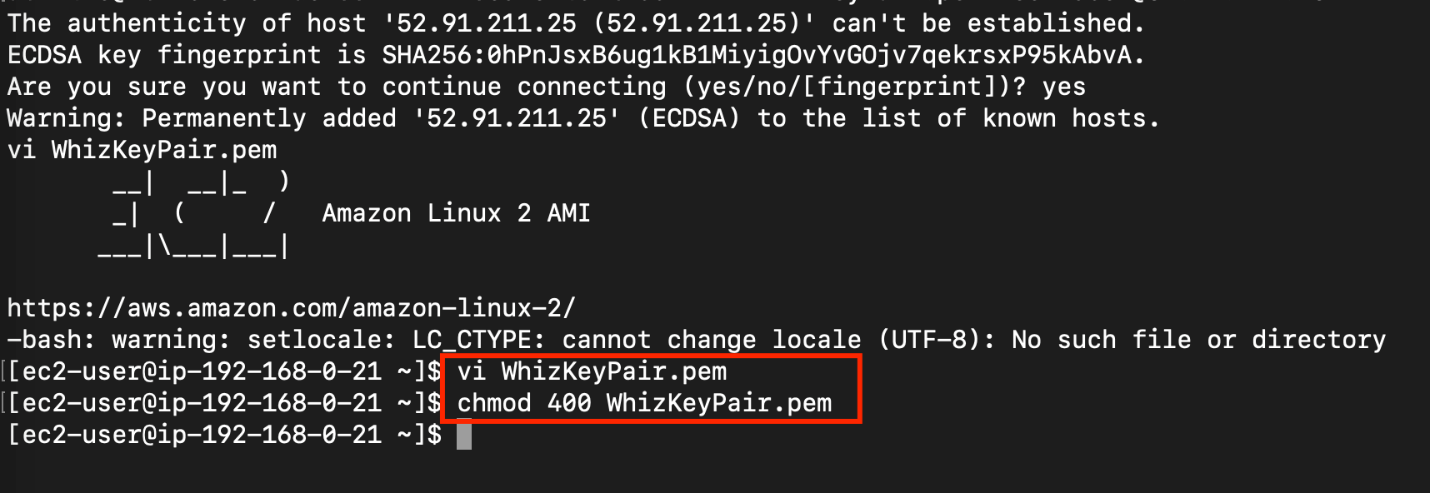
Task 12: SSH into Endpoint instance (Privately accessible) through Bastion host

1. [SSH into the Bastion instance](https://business.whizlabs.com/labs/support-document/ssh-into-ec-instance) using the Bastion PEM key: **WhizKeyPair.pem**
2. To SSH into **Endpoint instance via the Bastion instance**, we need the **WhizKeyPair.pem**to be present on the**Bastion instance.**
3. Open the **WhizKeyPair.pem** file on your local system and then**copy the text content**.
4. Navigate to the Bastion Instance and create a file named **WhizKeyPair.pem**using the below command:

vi WhizKeyPair.pem

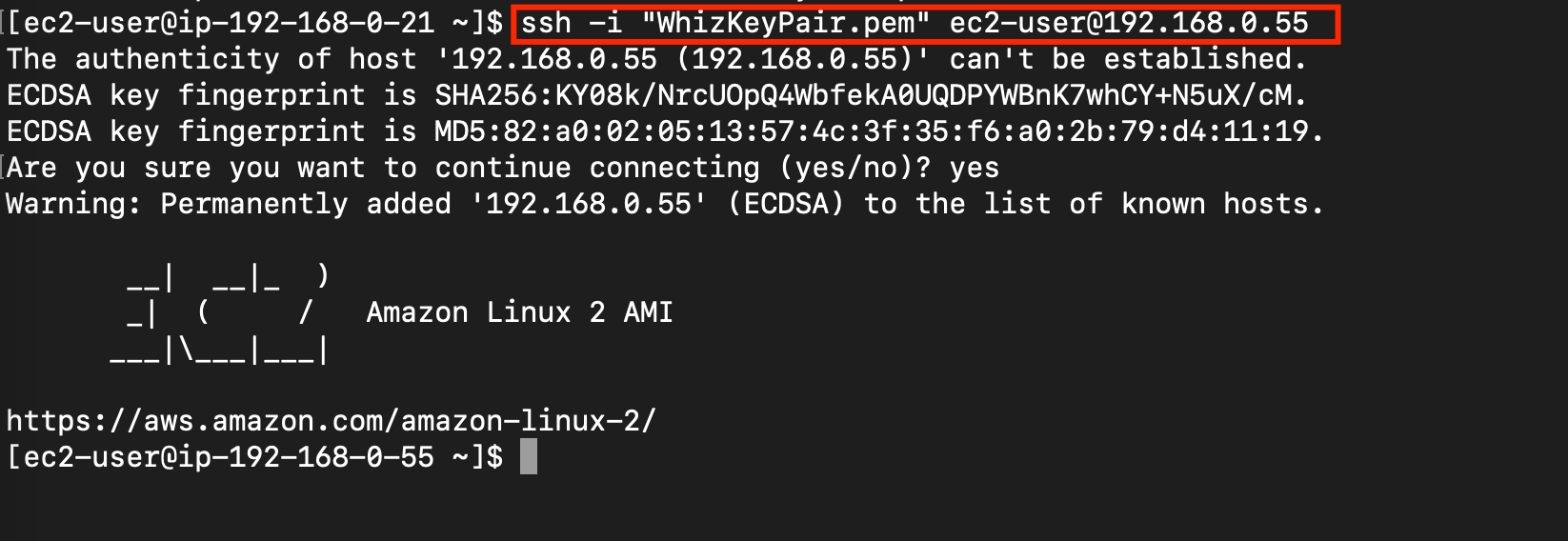
1. Paste the content and save it by pressing**shift+colon  followed by :wq!**and then enter to save your private key.
2. Make sure you have changed the **permission of the key file to 400**. You can change the permission using the below command:

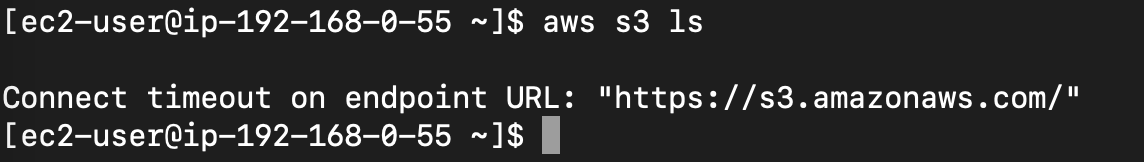
chmod 400 WhizKeyPair.pem

* ****

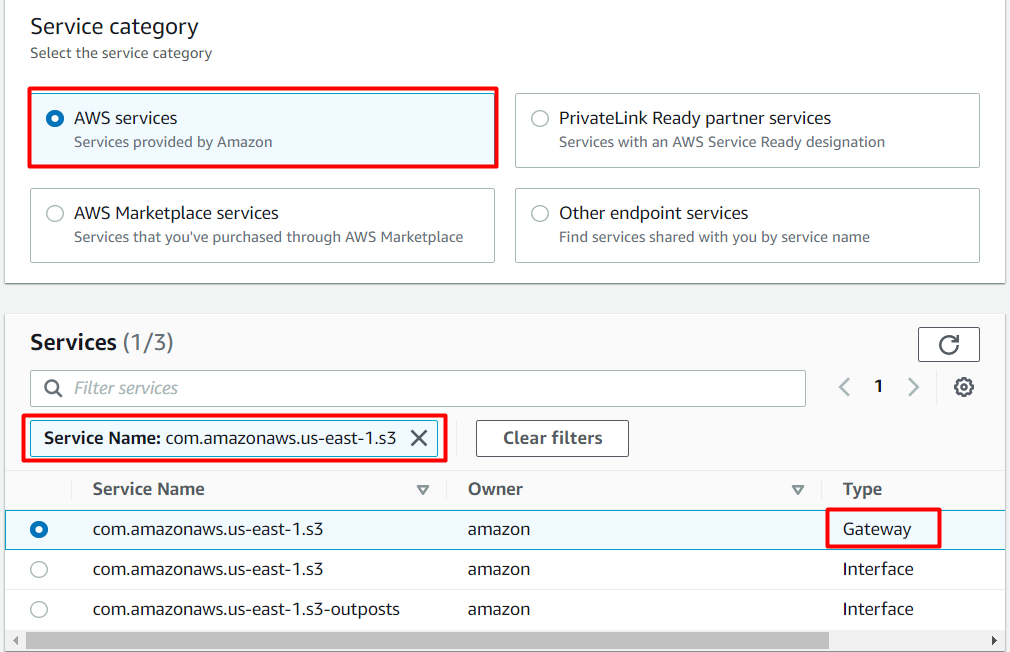
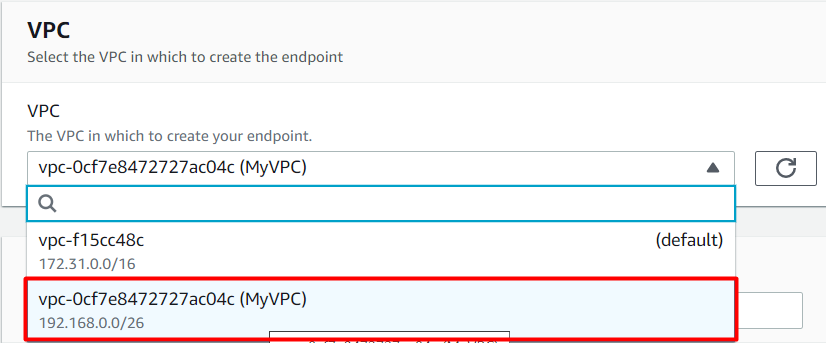
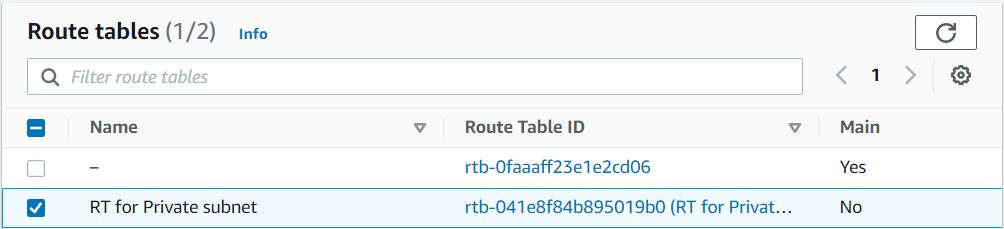
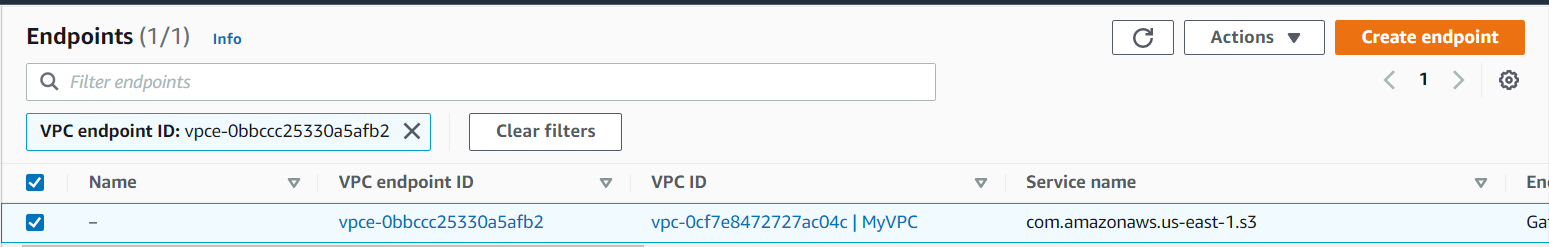
1. Now **you can log into the web servers**using the private key copied to the bastion server with the help of the below commands.

* **Note:** You **don't have public IP's** for the Endpoint instance since we created them in a private**subnet.**
* Syntax**: ssh -i WhizKeyPair.pem  ec2-user@<**Endpoint instance's Private IP**>**
* Example:**ssh -i WhizKeyPair.pem  ec2-user@192.168.0.55.**
* When asked for confirmation type: **yes**

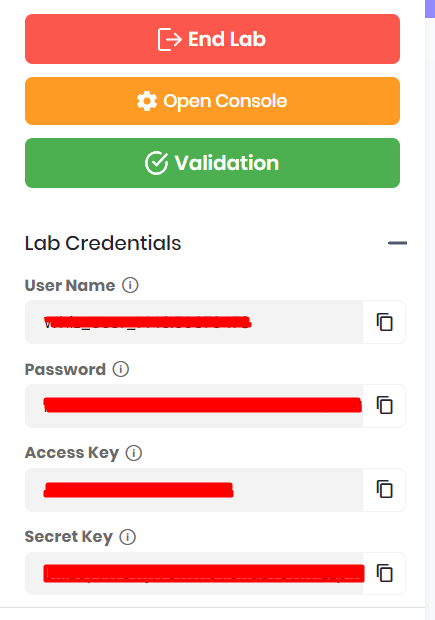


1. Though the assigned IAM role is having access for S3 Read, listing the bucket through AWS CLI command got failed, saying, connection timeout on S3's endpoint.  
   
2. As, this instance's security group is only allowed to do SSH, running any other command, will fail.
3. Let's add the permission to access the S3 endpoints using the VPC Endpoint for S3.

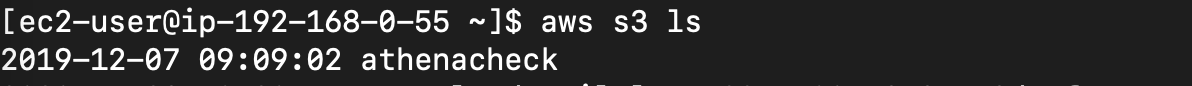
Task 13: Create a VPC Endpoint for S3, attach it to the Private subnet's Route table.

1. Navigate to **VPC** by clicking on the  menu at the top, then click on **VPC** in the **Networking and Content Delivery** section.
2. Click on   present in the **VIRTUAL PRIVATE CLOUD** section on the left sidebar.
3. Click on the 
4. Make sure the **Service category** is selected for **AWS services.** In the **Service name** search bar, **type *s3*,** and **press enter**  
     
   
5. The endpoint of **Type**, **Gateway** with Service name as **com.amazonaws.us-east-1.s3** will be listed.
6. Change the VPC, and select **MyVPC**.  
   
7. **Select the endpoint** and check the option for Route Table having name as **RT for Private subnet.**  
   
8. Finally, click on the 
9. An **endpoint** will be created.
10. Click on the **Close** button, and within a few moments, you will see the **endpoint** will be listed.  
    
11. (Optional) To check whether the endpoint is associated with the custom route table (RT for Private subnet) or not.  
    Go to the Route tables, Select the custom route table and click on the **Routes** options below, you will see an entry of S3.

Task 14: List all the S3 bucket and it's objects

1. Navigate back to your Bastion Instance and enter **aws configure.**
   * Access Key: Paste the access key provided to you.
   * Secret Key: Paste the secret key provided to you.
   * 
   * Default region name: us-east-1
   * Default output-format: Enter **[ENTER]**
2. List all the bucket's using the following AWS CLI command:

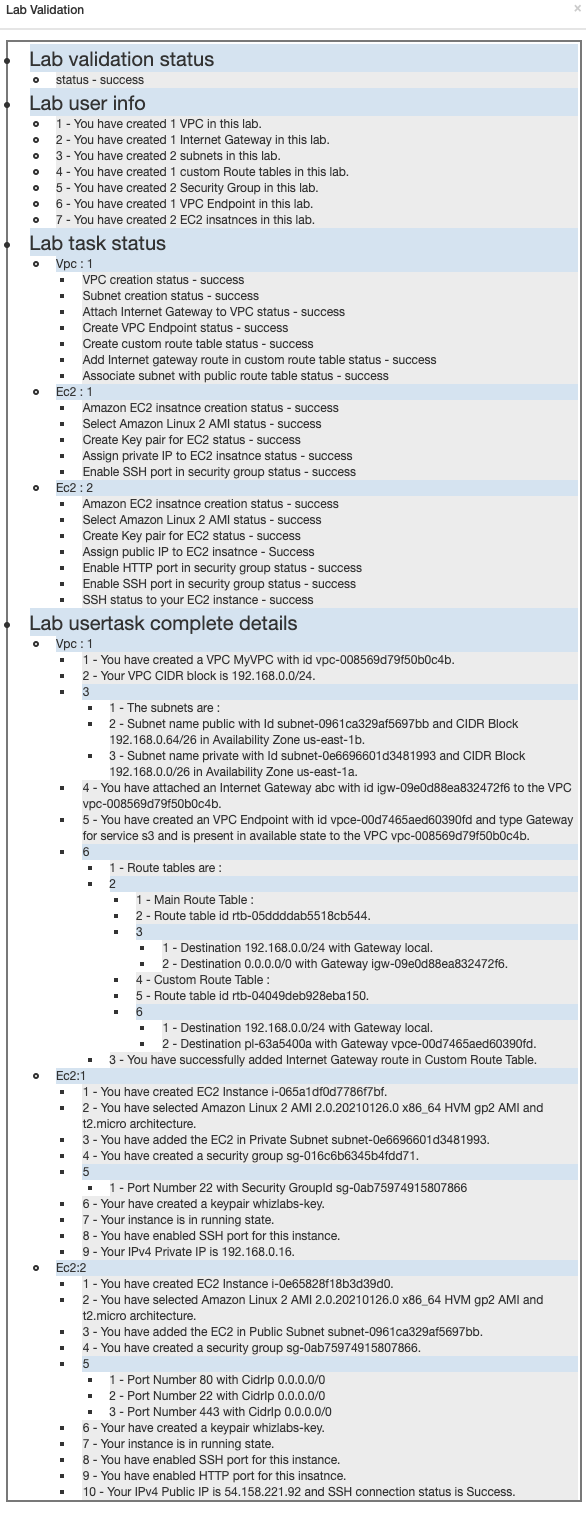
aws s3 ls

* + 

1. List the objects of the S3 bucket starting with name whizlabs..
   * Replace the S3 bucket name for the below command

aws s3 ls s3://whizlabs

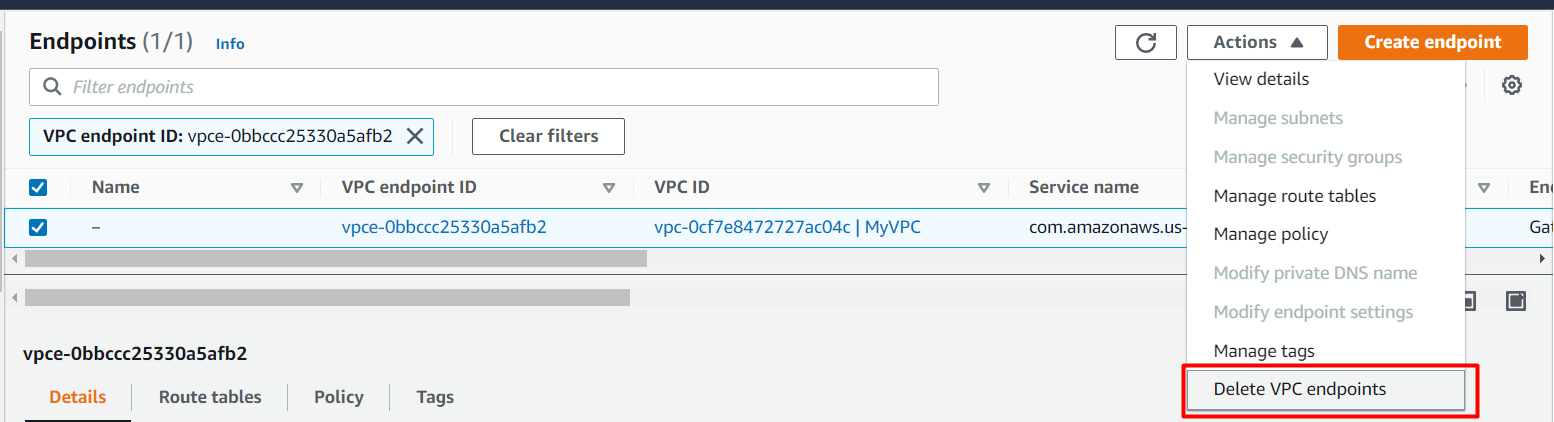
Task 15: Validation Test

1. Once the lab steps are completed, please click on the  button on the left side panel.
2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.
3. Sample output :   
   

Task 16: Delete AWS Resources

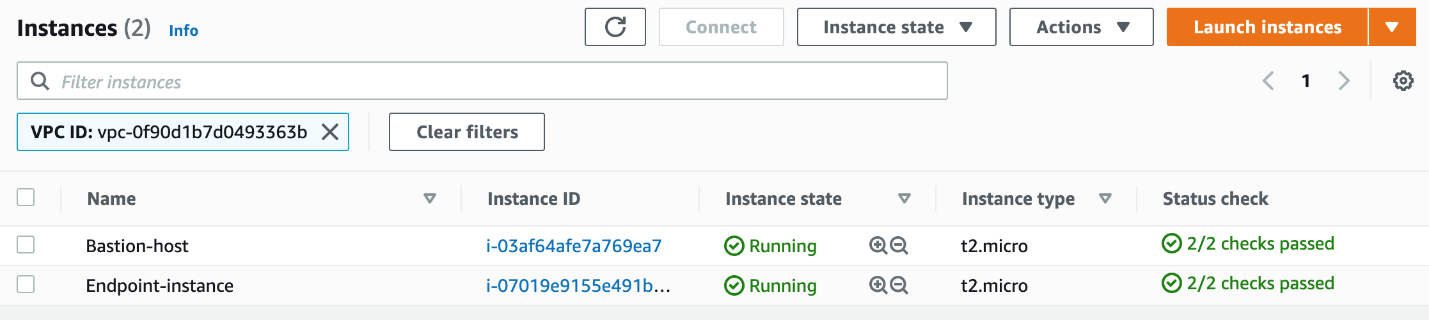
Delete VPC Endpoint

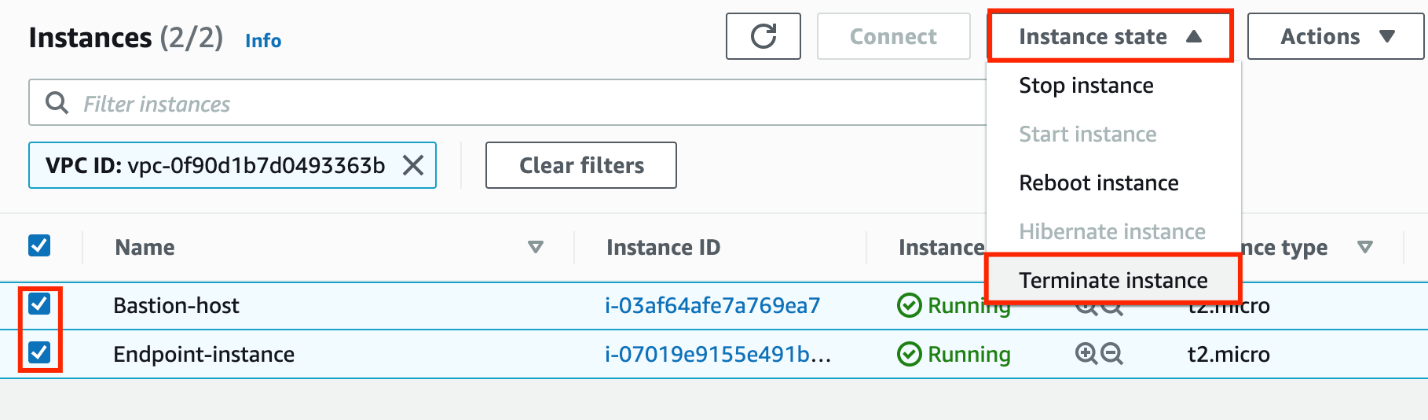
1. Navigate to **VPC** by clicking on the  menu at the top, then click on **VPC** in the **Networking and Content Delivery** section.
2. Click on  present in the **VIRTUAL PRIVATE CLOUD** section on the left sidebar.
3. To delete the VPC endpoint, perform the following tasks:
   * Select the Endpoint,
   * Click on the **Actions** button,
   * Choose the option of **Delete VPC endpoints**

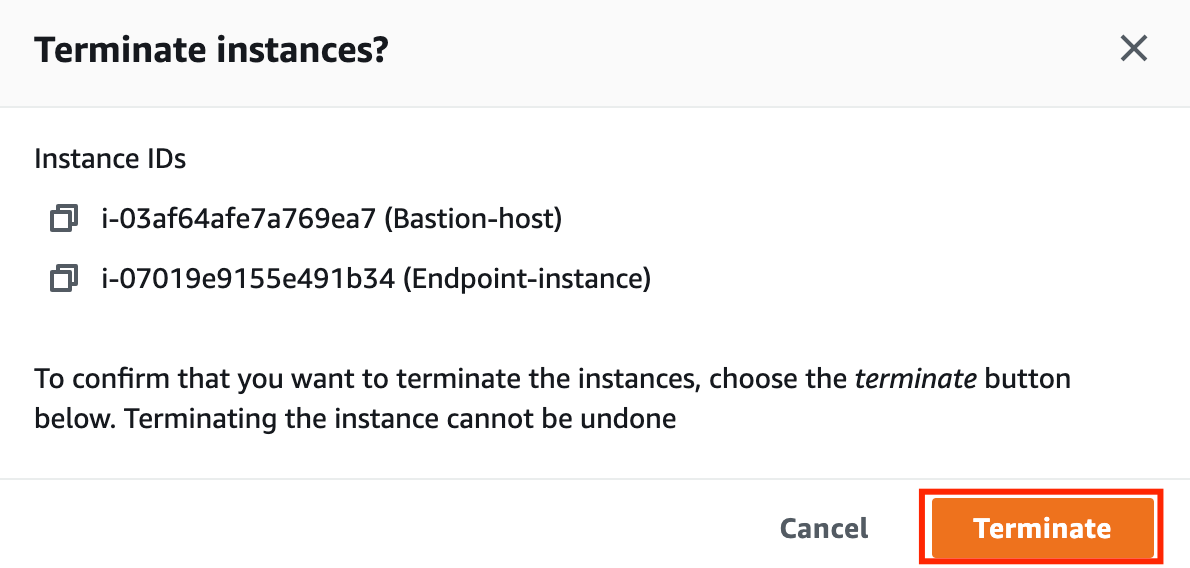


     4. Confirm the deletion by typing **delete.**  
     5. The endpoint will be deleted immediately.

EC2 Instance termination

1. Navigate to **EC2** by clicking on the  menu at the top, then click on **EC2** in the **Compute** section.
2. Click on **Instances**on the left panel.
3. EC2 Instances will be listed here.  
   
4. To terminate both the present instances, perform the following tasks:
   * Select both the **EC2 instances**
   * Click on the **Instance state**
   * Choose to **Terminate instance**



1. To confirm the termination of both the selected EC2 instance, click on the **Terminate** button.  
   
2. The instance will be terminated in a minute or so.  
   

**Completion and Conclusion**

1. We have launched two EC2 instances i.e. Bastion instance and Endpoint instance. We were able to SSH into the Endpoint instance via Bastion Instance successfully.
2. We have created a VPC endpoint for S3 to securely access S3 Buckets and their objects without going to the internet i.e. within Amazon's network through the Endpoint instance.
3. We tested the VPC endpoint for S3 from the private instance.

**End Lab**

1. Sign out of the AWS Account.
2. You have successfully completed the lab.
3. Once you have completed the steps, click on