Cloud Computing Lab

Experiment No.: 3

Create a Virtual Private Clouds and establish connections between each other. (Amazon VPC)

Experiment No. 3

Name : Sarvagya Singh

SAPID : 60009200030 **BATCH : K1**

1. **Aim*:*** Create a Virtual Private Clouds and establish connections between each other.

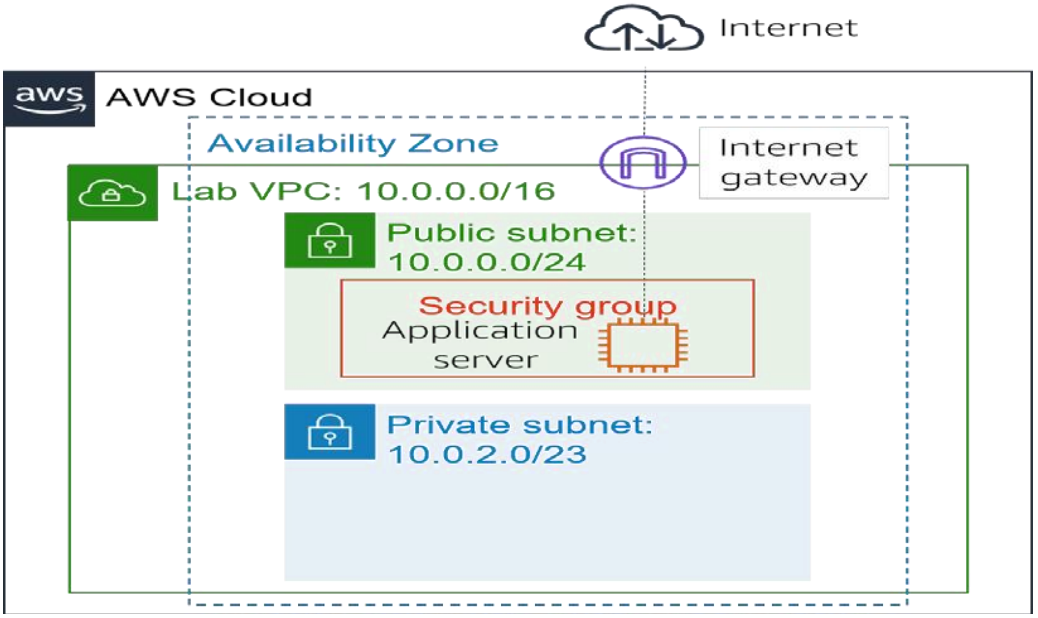
2. **Objectives: In this lab, you will explore**
    - To create your own VPC and add additional components to produce a customized network.
    - To create a security group.
    - Configure and customize an EC2 instance to run a web server and launch the EC2 instance to run in a subnet in the VPC.
3. **Outcomes: After completion of lab**
    - Deploy a VPC
    - Create an internet gateway and attach it to the VPC
    - Create a public subnet
    - Create a private subnet
    - Create an application server to test the VPC

4. Hardware / Software Required: Internet, AWS console

5. Theory:

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

In this lab students build the following infrastructure:


Internet
AWS Cloud
Availability Zone
Internet gateway
Lab VPC: 10.0.0.0/16
Public subnet: 10.0.0.0/24
Security group
Application server
Private subnet: 10.0.2.0/23

Task 1: Create Your VPC

A VPC is a virtual network that is dedicated to your Amazon Web Services (AWS) account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch AWS resources, such as Amazon Elastic Compute Cloud (Amazon EC2) instances, into the VPC. You can configure the VPC by modifying its IP address range and can create subnets. You can also configure route tables, network gateways, and security settings.

1. In the AWS Management Console, on the **Services** menu, choose **VPC**.

2. In the left navigation pane, choose **Your VPCs**.

   A default VPC is provided so that you can launch resources as soon as you start using AWS. There is also a shared VPC that you use later in the lab. However, you now create your own **Lab VPC**.

   The VPC will have a Classless Inter-Domain Routing (CIDR) range of **10.0.0.0/16**, which includes all IP address that start with 10.0.x.x. It contains more than 65,000 addresses. You later divide the addresses into separate subnets.

3. Choose **Create VPC** and configure the following settings:

   - For **Name tag**, enter Lab VPC
   - For **IPv4 CIDR block**, enter 10.0.0.0/1
   - Choose **Create VPC**.

4. From the **VPC Details** page, choose the **Tags** tab.
5. Choose **Actions** and select **Edit DNS hostnames**.

   This option assigns a friendly Domain Name System (DNS) name to EC2 instances in the VPC, such as the following:

   ec2-52-42-133-255.us-west-2.compute.amazonaws.com

6. Select **Enable**, and choose **Save changes**

Task 2: Creating subnets

A subnet is a subrange of IP addresses in the VPC. AWS resources can be launched into a specified subnet. Use a public subnet *for resources that must be connected to the internet, and use a* private subnet *for resources that must remain isolated from the internet.*

In this task, you create a public subnet and a private subnet shown in fig 7.2

Create a public subnet

7. In the left navigation pane, choose **Subnets**.
8. Choose **Create subnet** and configure the following settings:

- For **VPC ID**, choose **Lab VPC**.
- For **Subnet name**, enter Public Subnet
- For **Availability zone**, select the first Availability Zone in the list. Do not choose **No Preference**.
- For **IPv4 CIDR block**, enter 10.0.0.0/2
- Choose **Create subnet**

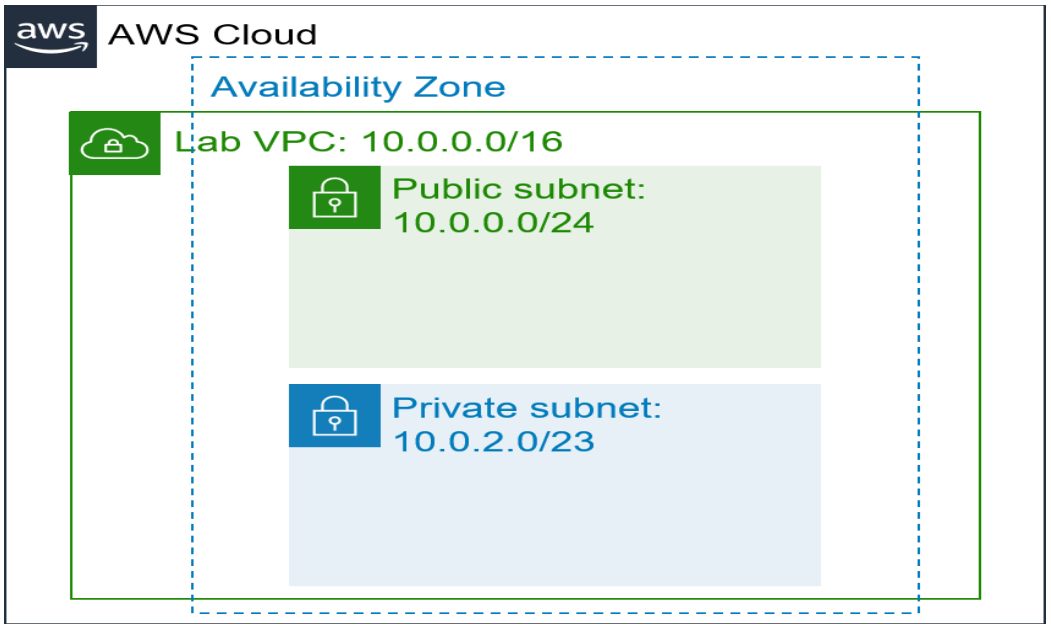


Figure 7.2 Public and Private Subnet

Create a private subnet

9. *Use what you learned in the previous steps to create another subnet with the following settings:*

- For **VPC ID**, choose **Lab VPC**.
- For **Subnet name,** enter Private Subnet
- For **Availability Zone**, select the first Availability Zone in the list. Do not choose **No Preference**.
- For **IPv4 CIDR block**, enter 10.0.2.0/2
- Choose **Create subnet**

VPC now has two subnets. However, the public subnet is totally isolated and cannot communicate with resources outside the VPC. Next, you configure the public subnet to connect to the internet via an internet gateway.

Task 3: Creating an internet gateway

10. In the left navigation pane, choose **Internet Gateways**.
11. Choose **Create internet gateway** and configure the following settings:

    - For **Name tag**, enter Lab IGW
    - Choose **Create internet gateway**

- *You can now attach the internet gateway to your **Lab VPC**.*

12. Choose **Actions** and then **Attach to VPC**, and configure the following settings:

    - For **Available VPCs**, select **Lab VPC**.
    - Choose **Attach internet gateway**

- This action attaches the internet gateway to your **Lab VPC** **Task**

4: Configuring route tables

13. In the left navigation pane, choose **Route Tables**.

*14. In the **VPC** column, find the route table that shows **Lab VPC**, and select the check box for this route table.*

15. In the **Name** column, choose and then enter the name Private Route Table *and choose* **Save**

16. In the lower half of the page, choose the **Routes** tab.

17. Choose **Create route table** and configure the following settings:

- For **Name**, enter Public Route Table
- For **VPC**, choose **Lab VPC**.
- Choose **Create route table**

18. In the **Routes** tab, choose **Edit routes**

You now add a route to direct internet-bound traffic (0.0.0.0/0) to the internet gateway.

19. Choose **Add route** and then configure the following settings:

    - For **Destination**, ente0.0.0.0/
    - For **Target**, select **Internet Gateway**, and then from the dropdown list select Lab IGW**.**
    - Choose **Save changes**

The last step associates this new route table with the public subnet.

20. Choose the **Subnet associations** tab.
21. In the **Subnets without explicit associations** section, choose **Edit subnet associations**
22. Select the row with **Public Subnet**.
23. Choose **Save associations**

Task 5: Creating a security group for the application server

24. In the left navigation pane, choose **Security Groups**.
25. Choose **Create security group** and configure the following settings:

    - For **Security group name**, enter App-SG
    - For **Description**, enterAllow HTTP traffic
    - For **VPC**, choose **Lab VPC**.
    - Choose **Create security group**

26. Choose the **Inbound Rules** tab.
27. Choose **Edit inbound rules**
28. *Choose **Add rule** and then configure the following settings:*

    - For **Type**, choose **HTTP**.
    - From the **Source type** dropdown list, choose **Anywhere IPv4**.
    - For **Description**, enter Allow web access
    - Choose **Save rules**

*You use this **App-SG** in the next task.*

Task 6: Launching an application server in the public subnet

To test that your VPC is correctly configured, you now launch an EC2 instance into the public subnet. You also confirm that you can access the EC2 instance from the internet.

35. On the **Services** menu, choose **EC2**.
36. Choose **Launch instance** and then select **Launch instance** from the dropdown list. Configure the following options:

- In the **Name and tags** pane, in the **Name** text box, enter App Server
- In the **Application and OS Images (Amazon Machine Image)** section, keep default selection, **Amazon Linux 2**.
- In the **Instance type** section, keep the default instance type, **t2.micro**.
- In the **Key pair (login)** section, from the **Key pair name - *required*** dropdown list, choose **Proceed without a key pair (not recommended)**.

In the **Network settings** section, choose **Edit**

- From the **VPC - *required*** dropdown list, choose **Lab VPC**.
- From the **Subnet** dropdown list, choose **Public Subnet**.
- Ensure that **Auto-assign public IP** is **Enable**.
- In the **Firewall (security groups)** section, choose **Select existing security group**
- From the **Common security groups** dropdown list, choose App-SG.
- In the **Configure storage** section, keep the default storage configuration.
- Expand the **Advanced details** section.
- For **IAM instance profile**, choose the role **Inventory-App-Role**.

- *Scroll down to **User data** section, copy and paste the below code in the block.*

```
#!/bin/bash
# Install Apache Web Server and PHP
yum install -y httpd mysql
amazon-linux-extras install -y php7.2
# Download Lab files
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod6-guided/scripts/inventory-app.zip
unzip inventory-app.zip -d /var/www/html/
# Download and install the AWS SDK for PHP
wget https://github.com/aws/aws-sdk-php/releases/download/3.62.3/aws.zip
unzip aws -d /var/www/html
```

- From the **Summary** section, choose **Launch instance**

37. Choose **View all instances**

38. *Wait for the application server to fully launch. It should display the following status:*

- **Instance State:** Running

39. Select **App Server**.
40. From the **Details** tab, copy the **Public IPv4 address** address.
41. *Open a new browser tab, paste the IP address you just copied, and press Enter.*

If you configured the VPC correctly, the Inventory application and this message should appear: **Please configure Settings to connect to database**.

6. Result : Paste your screen shots for every task.

7. Conclusions :

With Amazon Virtual Private Cloud (Amazon VPC), you can launch AWS resources in a logically isolated virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

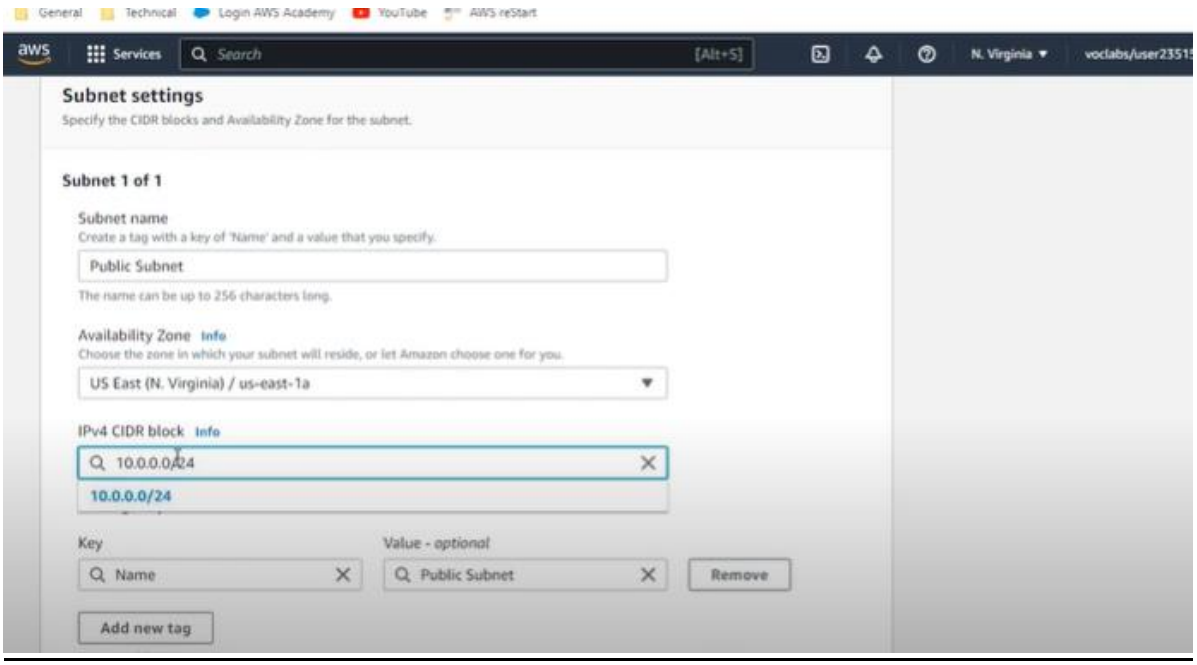
The following diagram shows an example VPC. The VPC has one subnet in each of the Availability Zones in the Region, EC2 instances in each subnet, and an internet gateway to allow communication between the resources in your VPC and the internet.

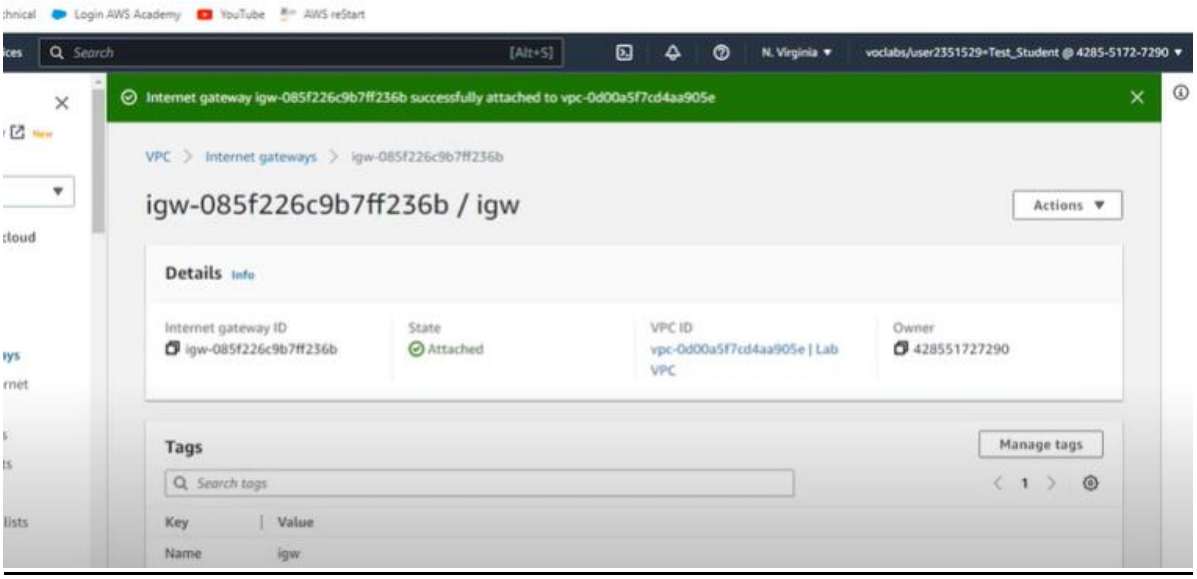
8. Viva Questions:

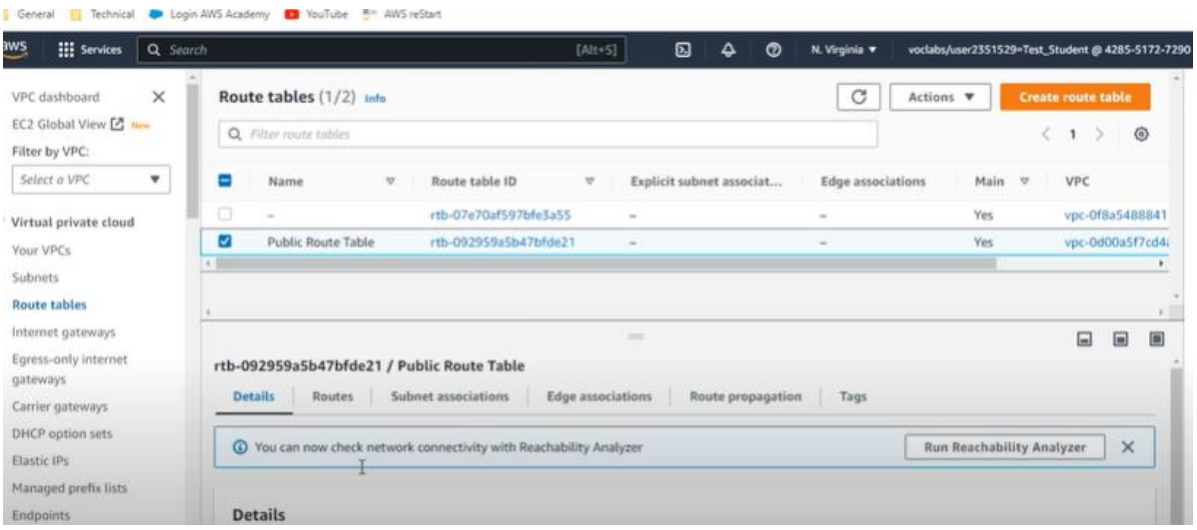
- What are the components of Amazon VPC
- What are Internet Gateways in VPC?
- What do you know about VPC Peering?

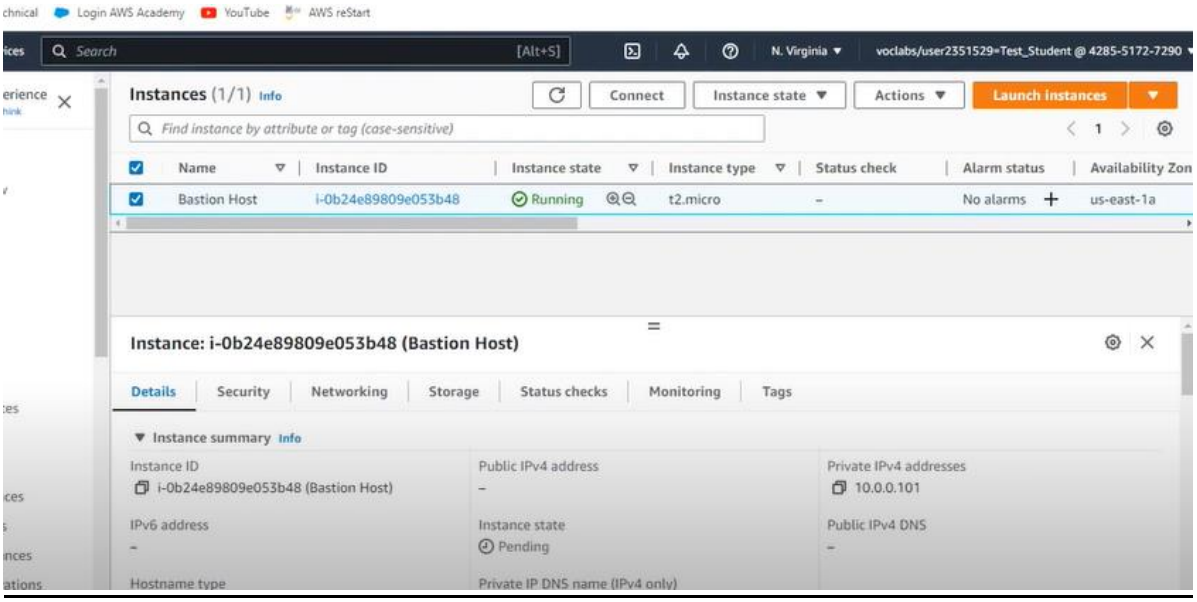
References: https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html

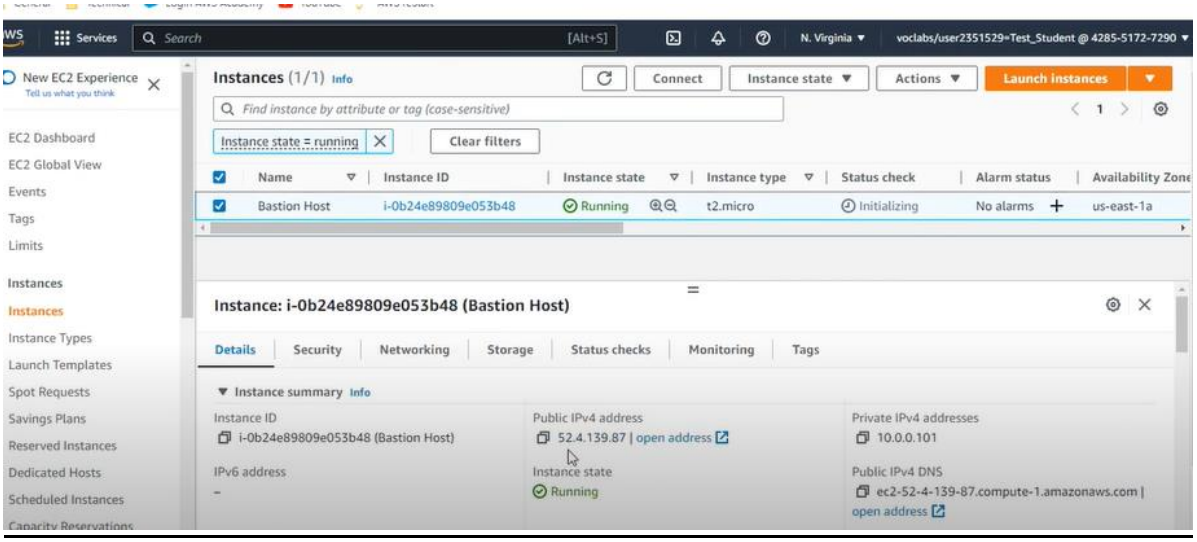
9. Screenshots :

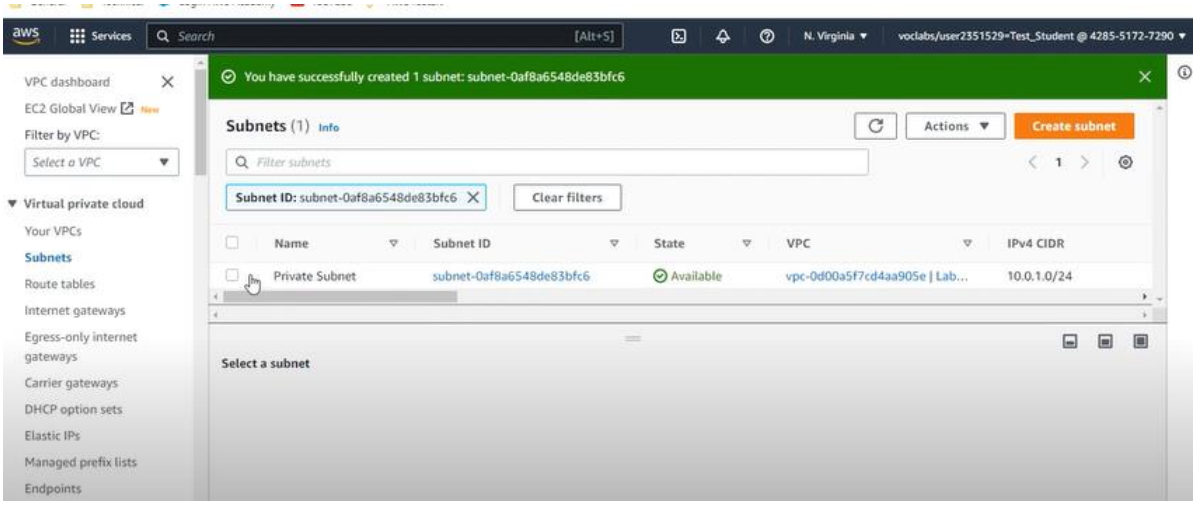

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.
Subnet 1 of 1
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
Public Subnet
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1a
IPv4 CIDR block Info
10.0.0.0/24
Key
Name
Value - optional
Public Subnet
Remove
Add new tag

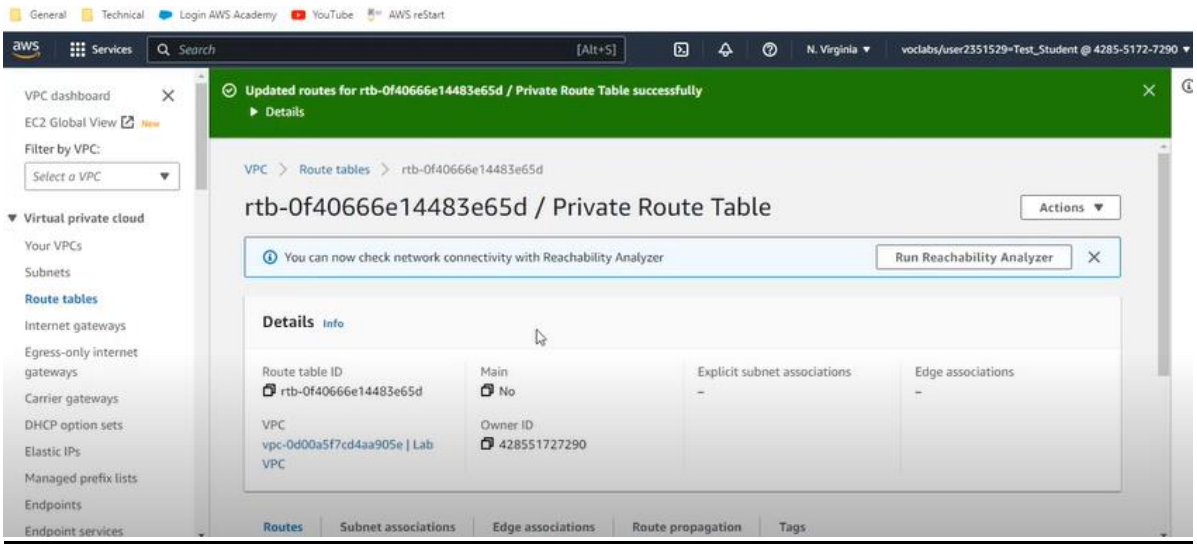

Internet gateway igw-085f226c9b7ff236b successfully attached to vpc-0d00a5f7cd4aa905e
VPC > Internet gateways > igw-085f226c9b7ff236b
igw-085f226c9b7ff236b / igw
Actions
Details Info
Internet gateway ID
igw-085f226c9b7ff236b
State
Attached
VPC ID
vpc-0d00a5f7cd4aa905e | Lab VPC
Owner
428551727290
Tags
Manage tags
Search tags
Key
Value
Name
igw

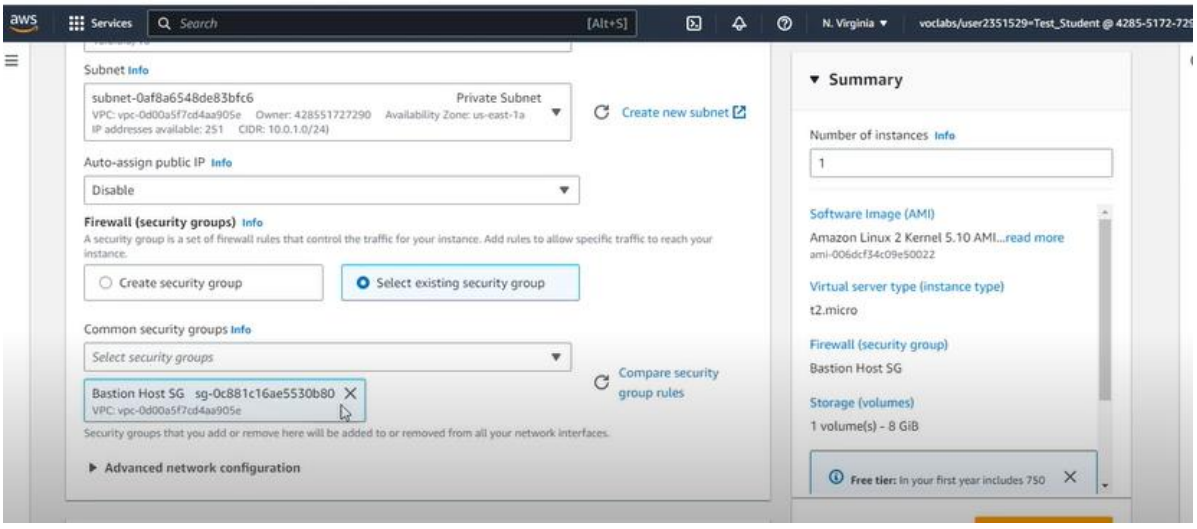

VPC dashboard
EC2 Global View
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Route tables (1/2) Info
Actions
Create route table
Filter route tables
Name
Route table ID
Explicit subnet associat...
Edge associations
Main
VPC
rtb-07e70af597bfe3a55
Yes
Public Route Table
rtb-092959a5b47bfde21
Yes
rtb-092959a5b47bfde21 / Public Route Table
Details
Routes
Subnet associations
Edge associations
Route propagation
Tags
You can now check network connectivity with Reachability Analyzer
Run Reachability Analyzer
Details

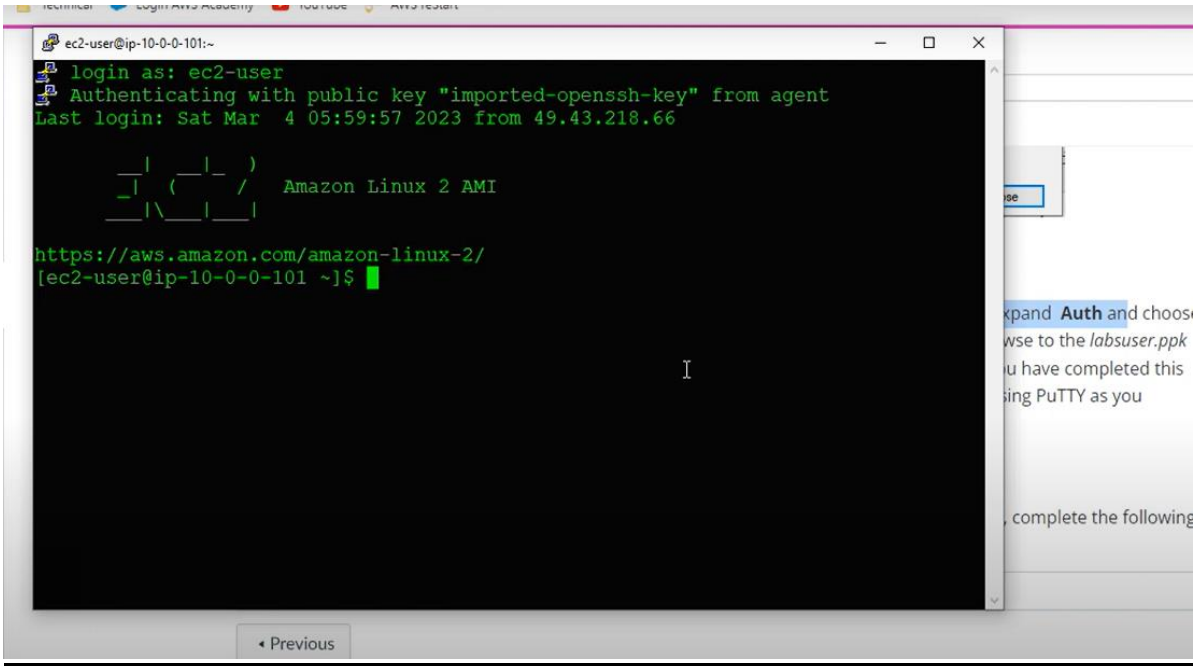

Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zon
Bastion Host
i-0b24e89809e053b48
Running
t2.micro
No alarms
us-east-1a
Instance: i-0b24e89809e053b48 (Bastion Host)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-0b24e89809e053b48 (Bastion Host)
Public IPv4 address
Private IPv4 addresses
10.0.0.101
IPv6 address
Instance state
Pending
Public IPv4 DNS
Hostname type
Private IP DNS name (IPv4 only)

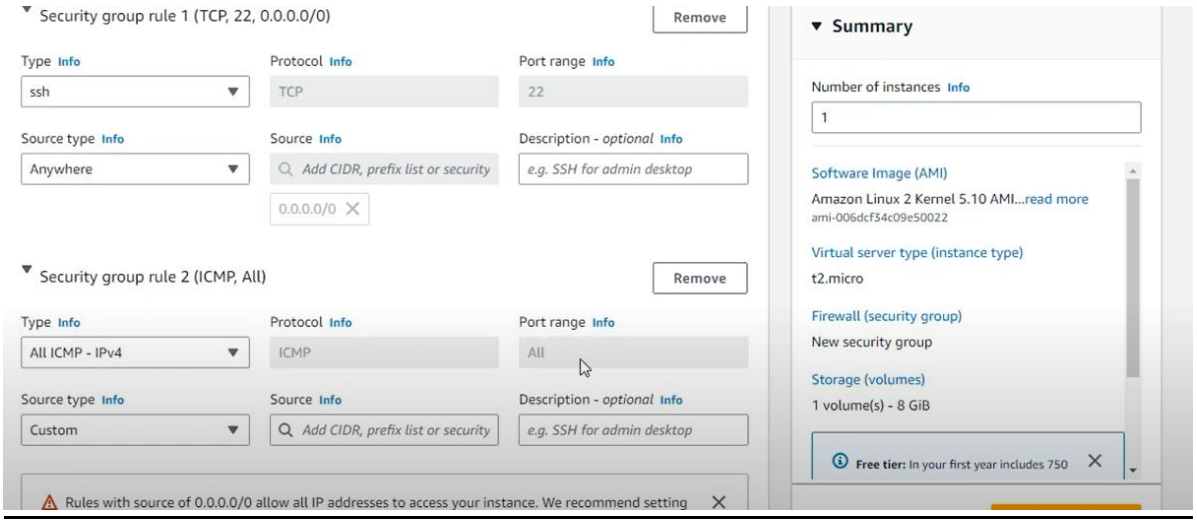

New EC2 Experience
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Scheduled Instances
Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Find instance by attribute or tag (case-sensitive)
Instance state = running
Clear filters
Bastion Host
i-0b24e89809e053b48
Running
t2.micro
Initializing
No alarms
us-east-1a
Instance: i-0b24e89809e053b48 (Bastion Host)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-0b24e89809e053b48 (Bastion Host)
Public IPv4 address
52.4.139.87 | open address
Private IPv4 addresses
10.0.0.101
IPv6 address
Instance state
Running
Public IPv4 DNS
ec2-52-4-139-87.compute-1.amazonaws.com | open address


VPC dashboard
EC2 Global View
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
You have successfully created 1 subnet: subnet-0af8a6548de83bfc6
Subnets (1) Info
Actions
Create subnet
Filter subnets
Subnet ID: subnet-0af8a6548de83bfc6
Clear filters
Name
Subnet ID
State
VPC
IPv4 CIDR
Private Subnet
subnet-0af8a6548de83bfc6
Available
vpc-0d00a5f7cd4aa905e | Lab...
10.0.1.0/24
Select a subnet


Updated routes for rtb-0f40666e14483e65d / Private Route Table successfully
rtb-0f40666e14483e65d / Private Route Table
Actions
You can now check network connectivity with Reachability Analyzer
Run Reachability Analyzer
Details
Route table ID
rtb-0f40666e14483e65d
Main
No
Explicit subnet associations
Edge associations
VPC
vpc-0d00a5f7cd4aa905e | Lab VPC
Owner ID
428551727290
Routes
Subnet associations
Edge associations
Route propagation
Tags


Subnet
subnet-0af8a6548de83bfc6
Private Subnet
Create new subnet
Auto-assign public IP
Disable
Firewall (security groups)
Create security group
Select existing security group
Common security groups
Select security groups
Bastion Host SG sg-0c881c16ae5530b80
Compare security group rules
Advanced network configuration
Summary
Number of instances
1
Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
Virtual server type (instance type)
t2.micro
Firewall (security group)
Bastion Host SG
Storage (volumes)
1 volume(s) - 8 GiB


ec2-user@ip-10-0-0-101:~
login as: ec2-user
Authenticating with public key "imported-openssh-key" from agent
Last login: Sat Mar 4 05:59:57 2023 from 49.43.218.66
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-101 ~]$
Previous


Security group rule 1 (TCP, 22, 0.0.0.0/0)
Remove
Type Info
ssh
Protocol Info
TCP
Port range Info
22
Source type Info
Anywhere
Source Info
Add CIDR, prefix list or security
0.0.0.0/0
Description - optional Info
e.g. SSH for admin desktop
Security group rule 2 (ICMP, All)
Remove
Type Info
All ICMP - IPv4
Protocol Info
ICMP
Port range Info
All
Source type Info
Custom
Source Info
Add CIDR, prefix list or security
Description - optional Info
e.g. SSH for admin desktop
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-006dcf34c09e50022
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750