Lesson 10 Demo 04

Dynamic Port Integration with Application Load Balancer

Objective: To enable dynamic port mapping in a container

Tools required: AWS account

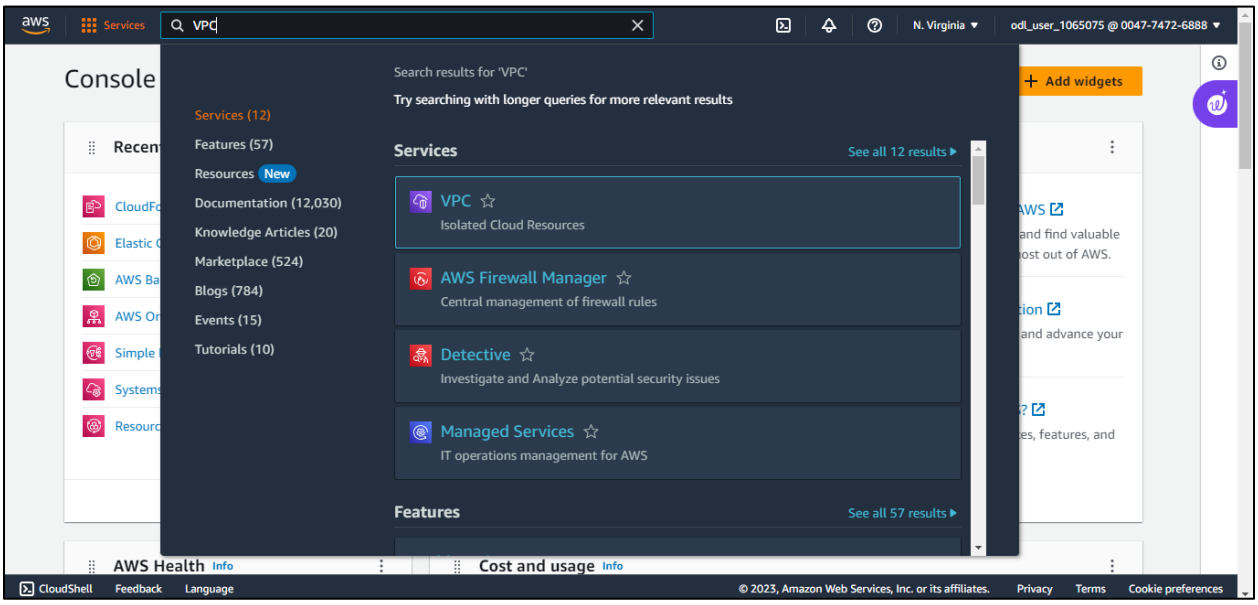
Prerequisites: NA

Steps to be followed:

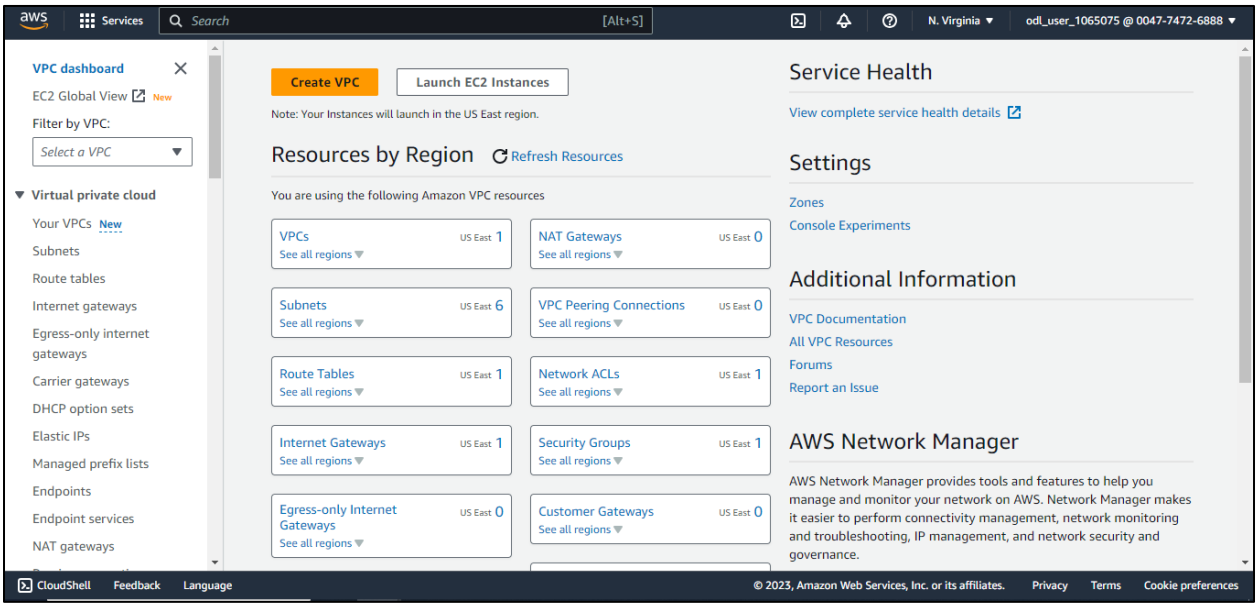
1. Create a custom VPC and enable the DNS hostname
2. Create an Internet Gateway
3. Create three subnets
4. Create a route table and attach it to three subnets
5. Create a cluster
6. Create a task definition
7. Run the service on the cluster

Step 1: Create a custom VPC and enable the DNS hostname

1.1 In the AWS Management Console, search for and click **VPC**

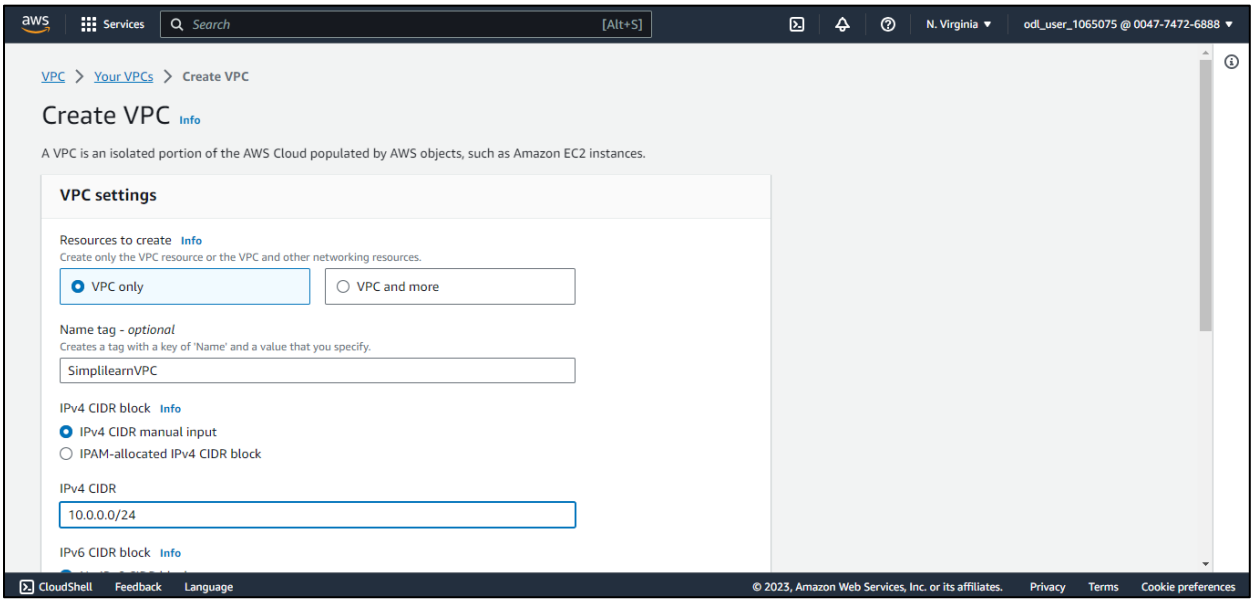


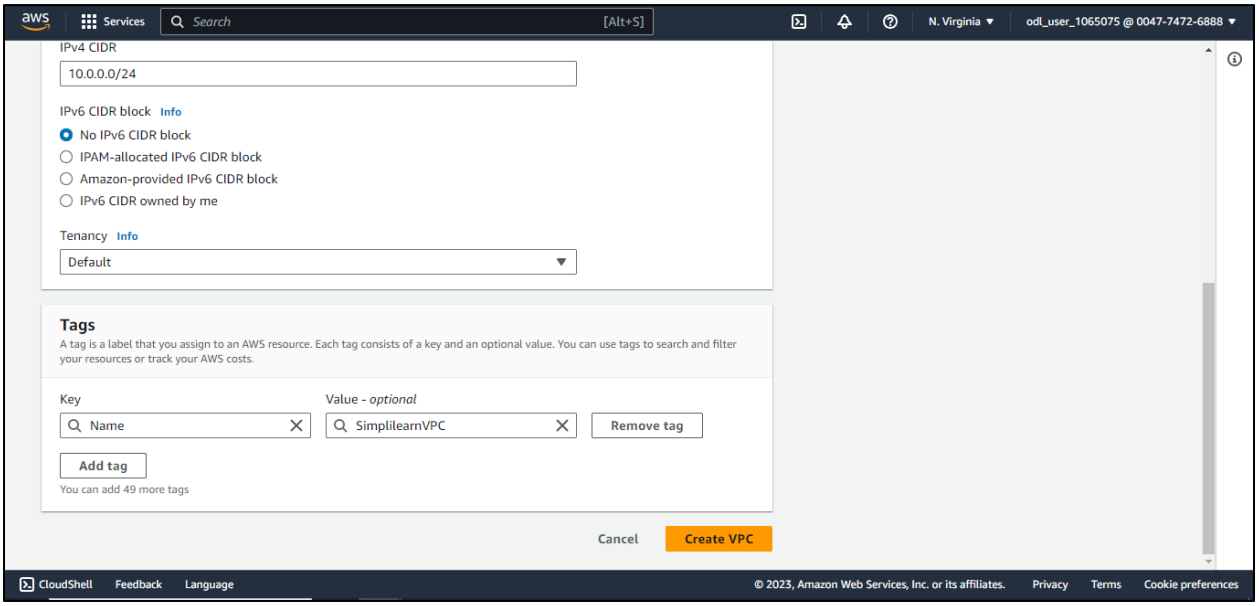
1.2 In the VPC dashboard, click on **Create VPC**



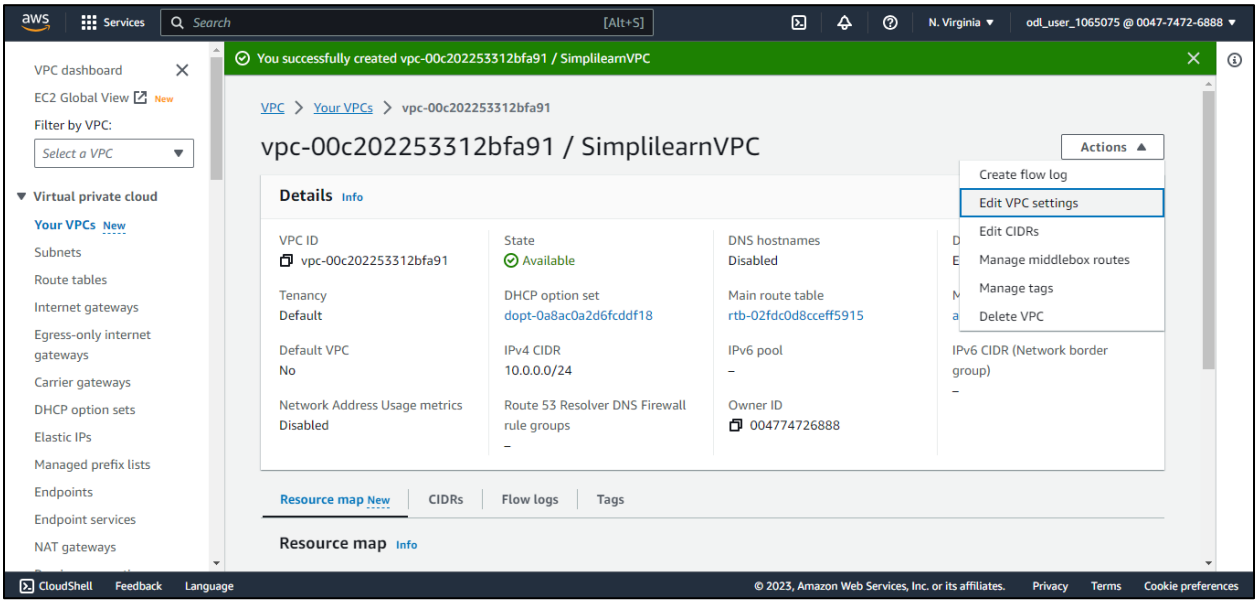
1.3 In the **Create VPC** window, perform the following:

- Select **VPC only** under the **Resource to create** section
- Enter an arbitrary name for the VPC under the **Name tag** section
- Enter **10.0.0.0/24** in the **IPv4 CDR** and click on **Create VPC**

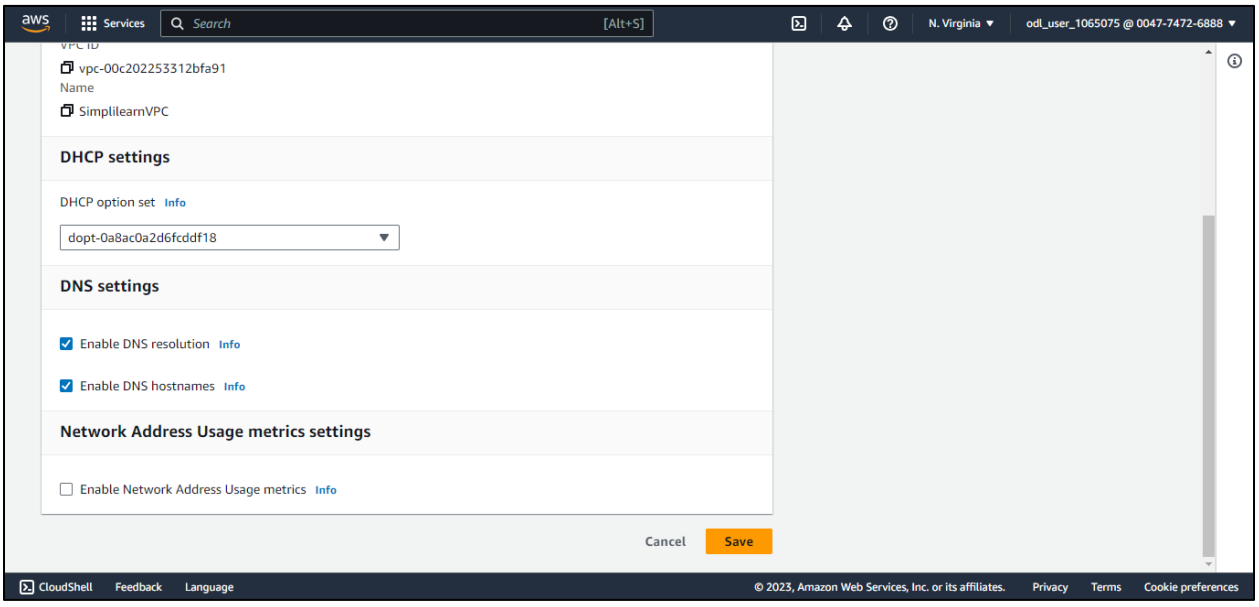




1.4 In the **VPC dashboard,** click on the **Actions** dropdown menu and select **Edit VPC Settings**

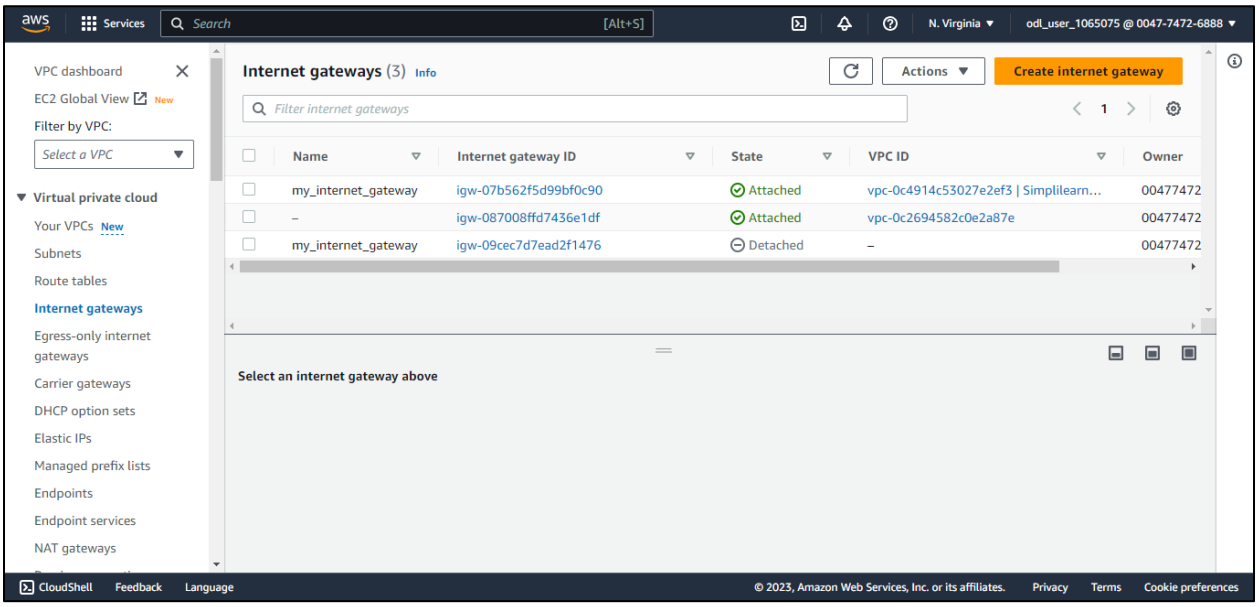


1.5 Check the **Enable DNS hostnames** checkbox and click on **Save**

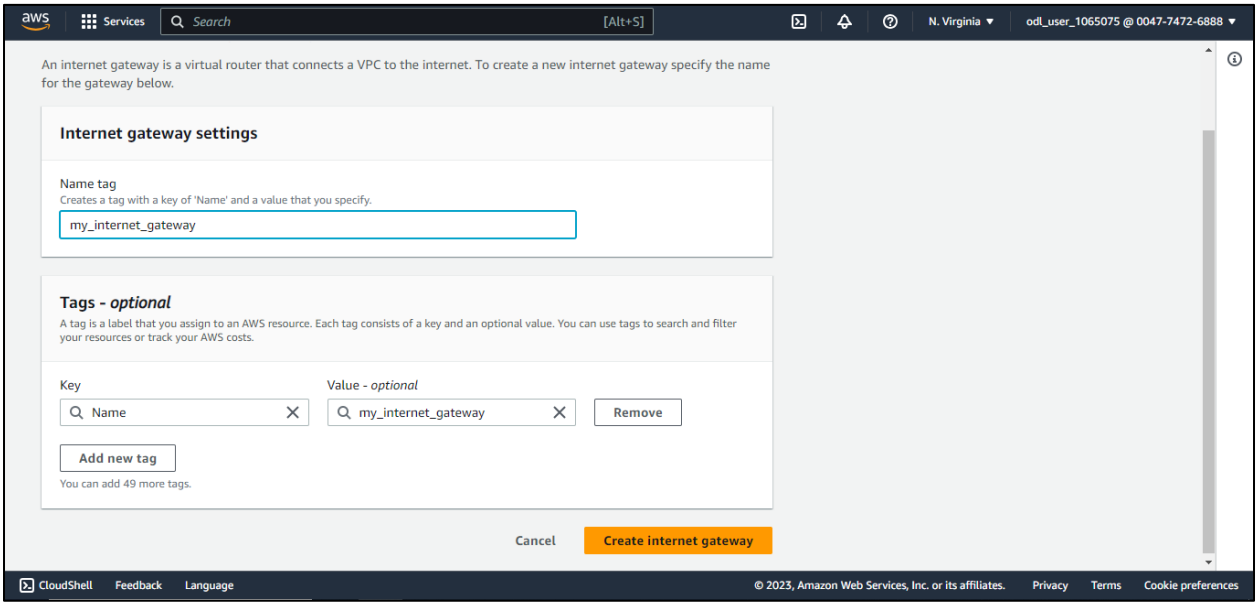


Step 2: Create an Internet gateway

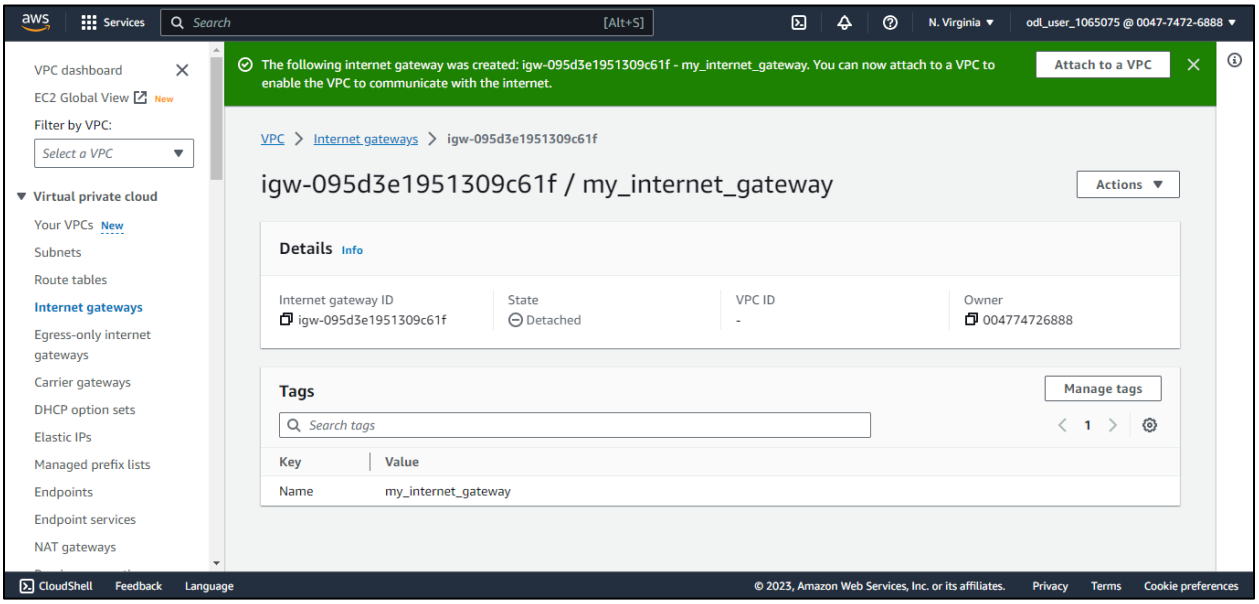
2.1 Open the **VPC dashboard** and click on **Create internet gateway**



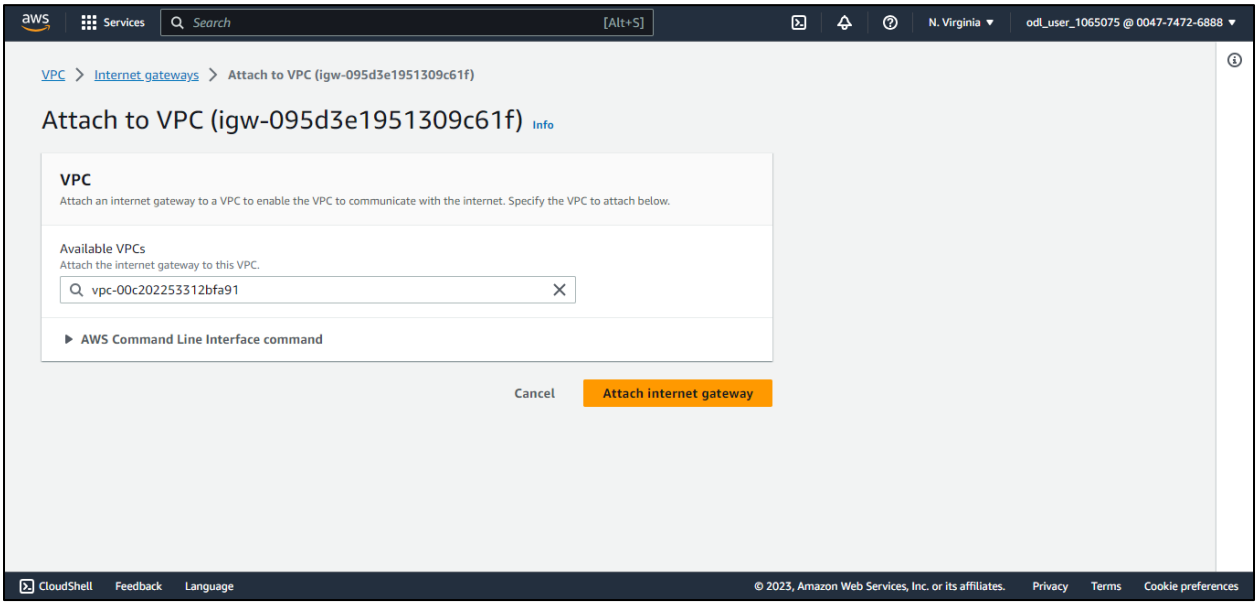
2.2 Enter an arbitrary name for the **Internet gateway settings** in the **Name tag** and click on **Create internet gateway**



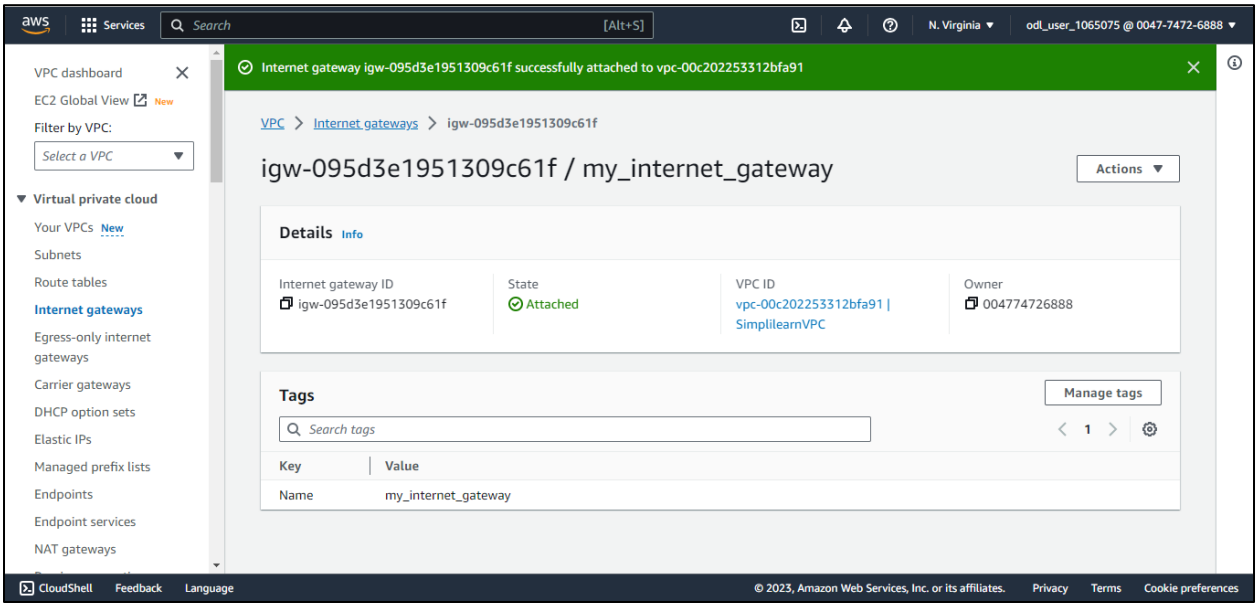
2.3 In the **Internet gateways dashboard**, click on **Attach to a VPC**



2.4 In the **Available VPCs**, select the VPC created in step 1 and click on **Attach internet gateway**

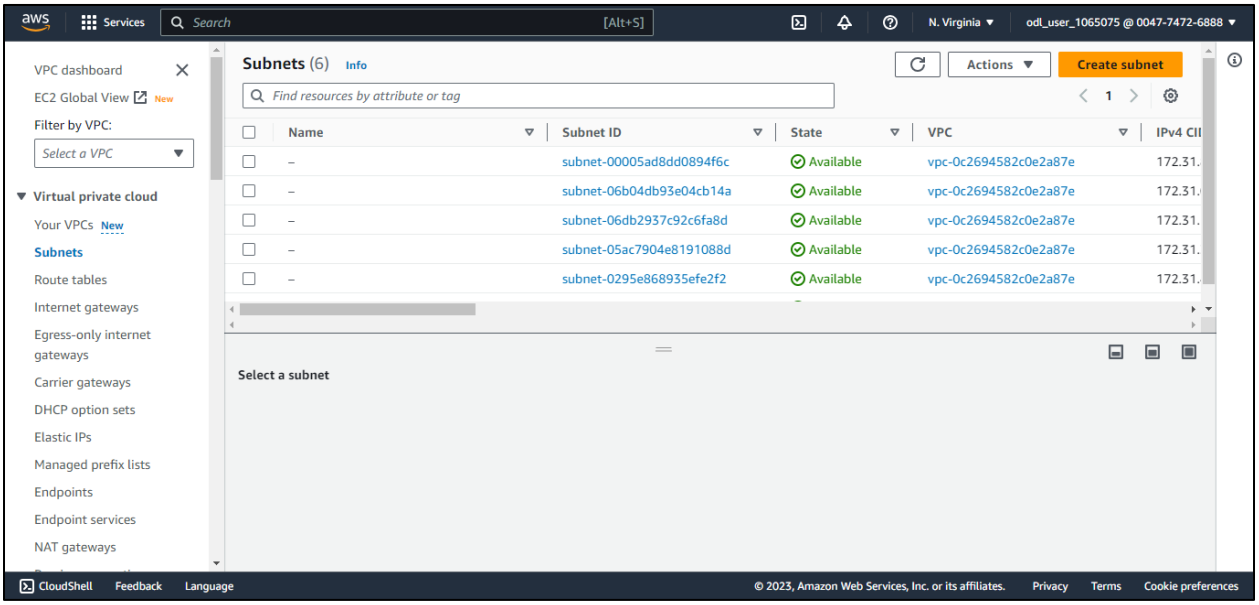


The following screen will appear:

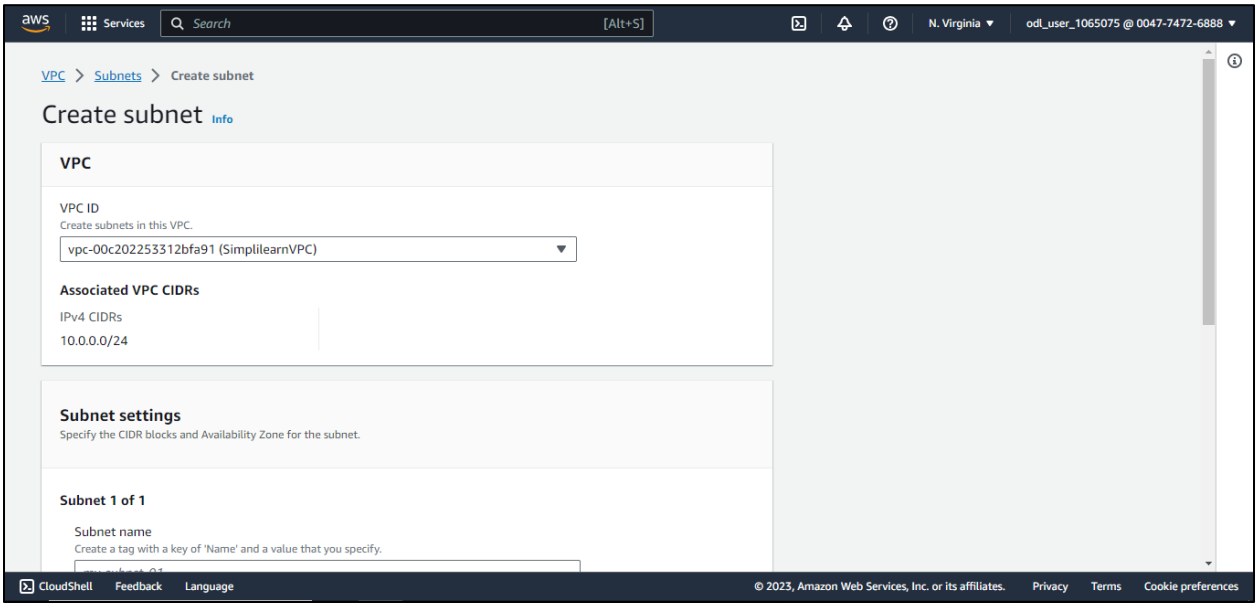


Step 3: Create three subnets

3.1 Click on **Subnets** and click on **Create subnet**

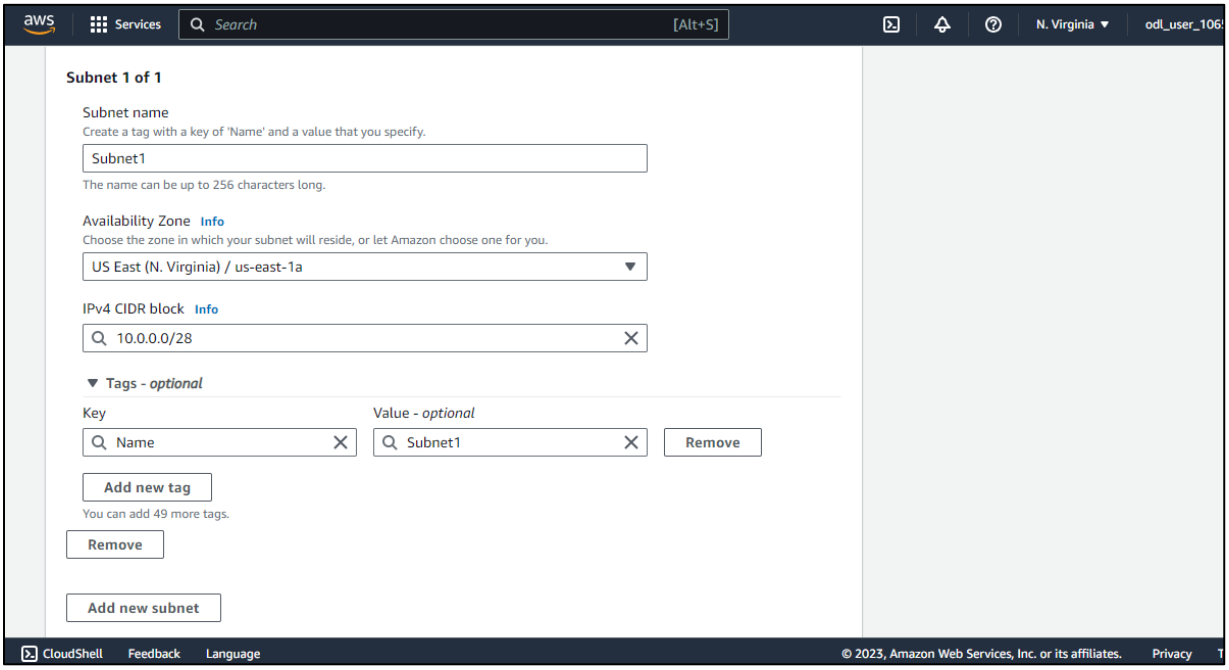


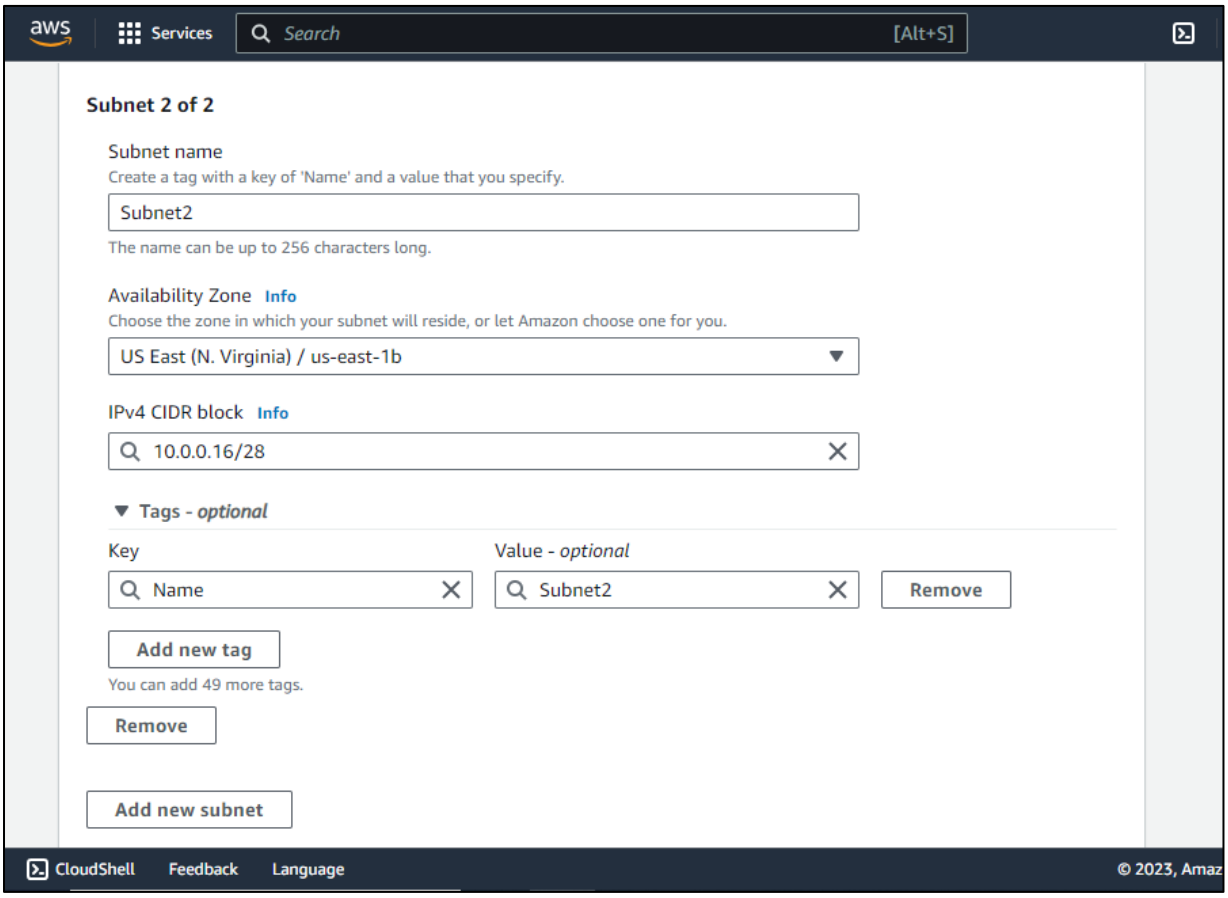
3.2 In the **VPC ID**, enter the VPC created in step 1

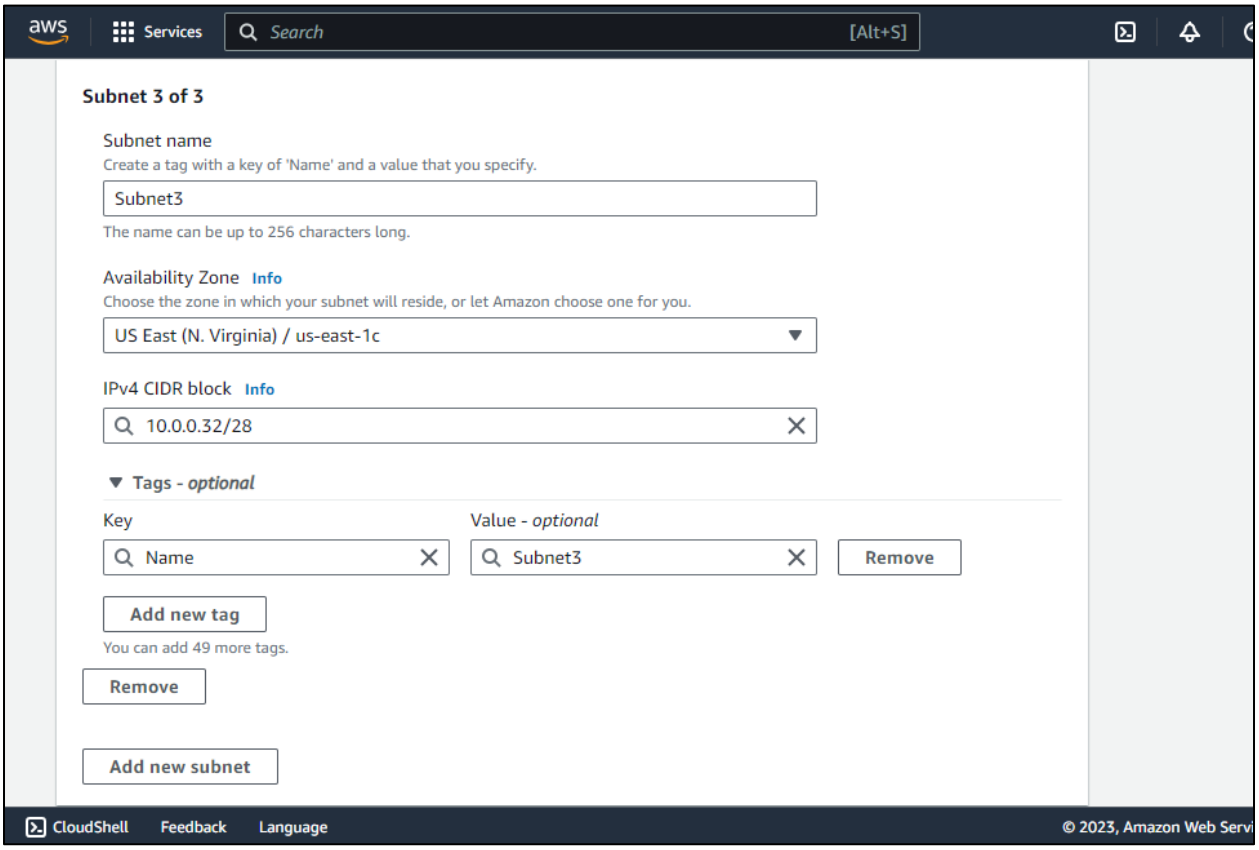


3.3 In the **Subnet settings** section, repeat steps 3.1 and 3.2 with the following settings:

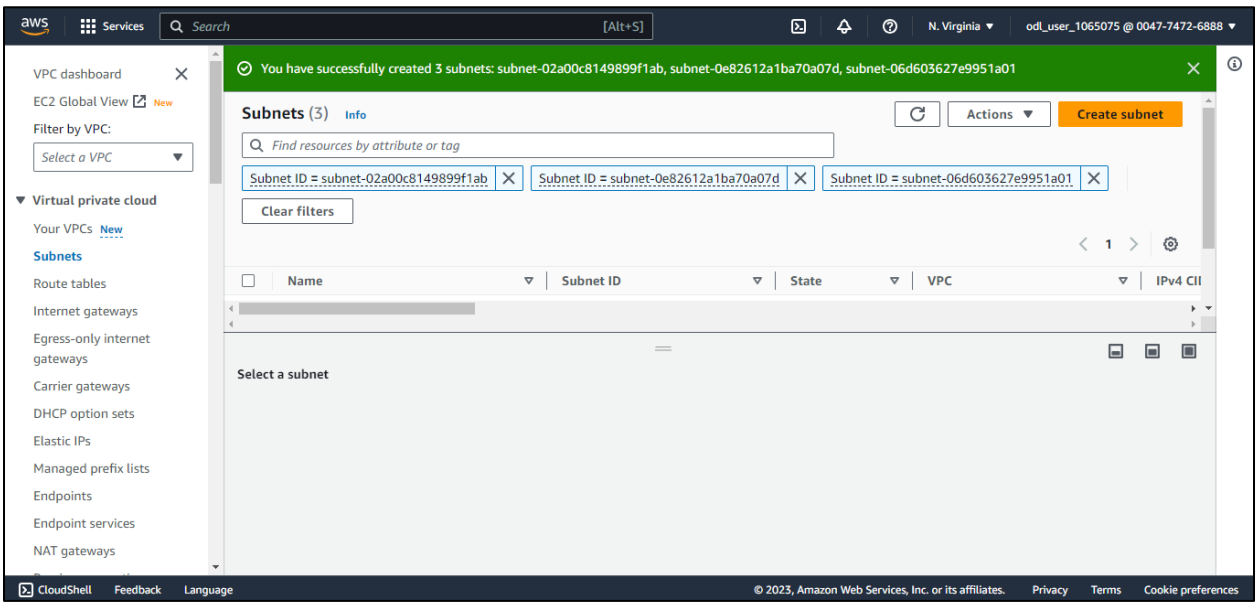
- Create a subnet in the **us-east-1a** zone and enter **10.0.0.0/28** in the **IPv4 CIDR block**
- Create a subnet in the **us-east-1b** zone and enter **10.0.0.16/28** in the **IPv4 CIDR block**
- Create a subnet in the **us-east-1c** zone and enter **10.0.0.32/28** in the **IPv4 CIDR block**



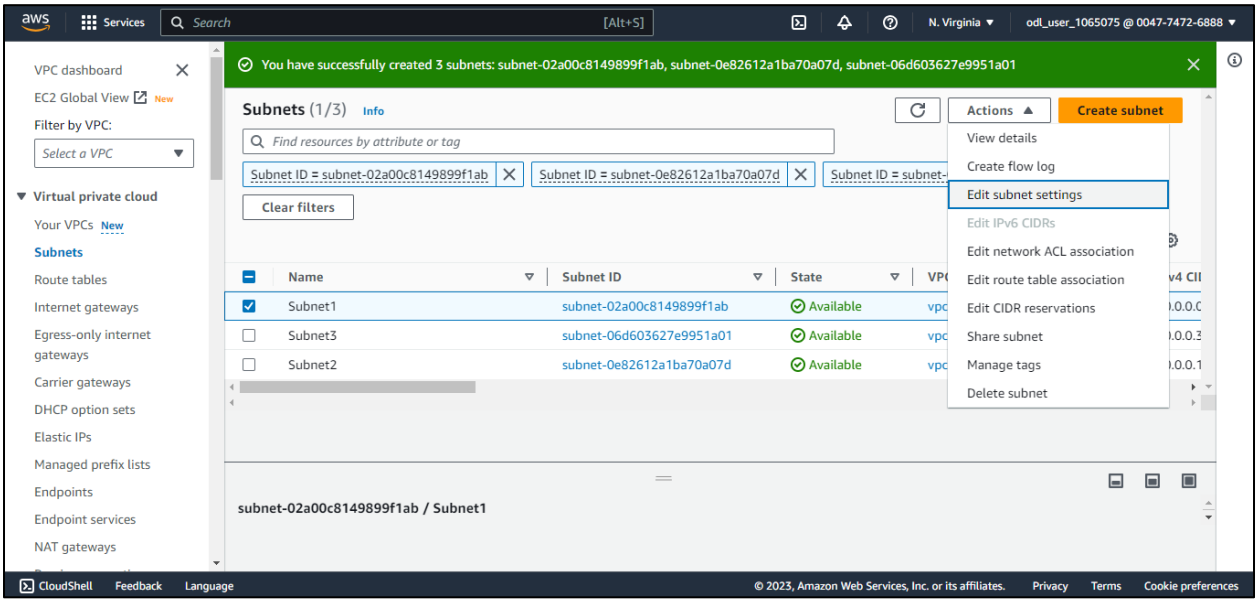

aws
Services
Search
[Alt+S]
Subnet 2 of 2
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
Subnet2
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1b
IPv4 CIDR block Info
10.0.0.16/28
Tags - optional
Key
Value - optional
Name
Subnet2
Remove
Add new tag
You can add 49 more tags.
Remove
Add new subnet
CloudShell
Feedback
Language
© 2023, Amaz



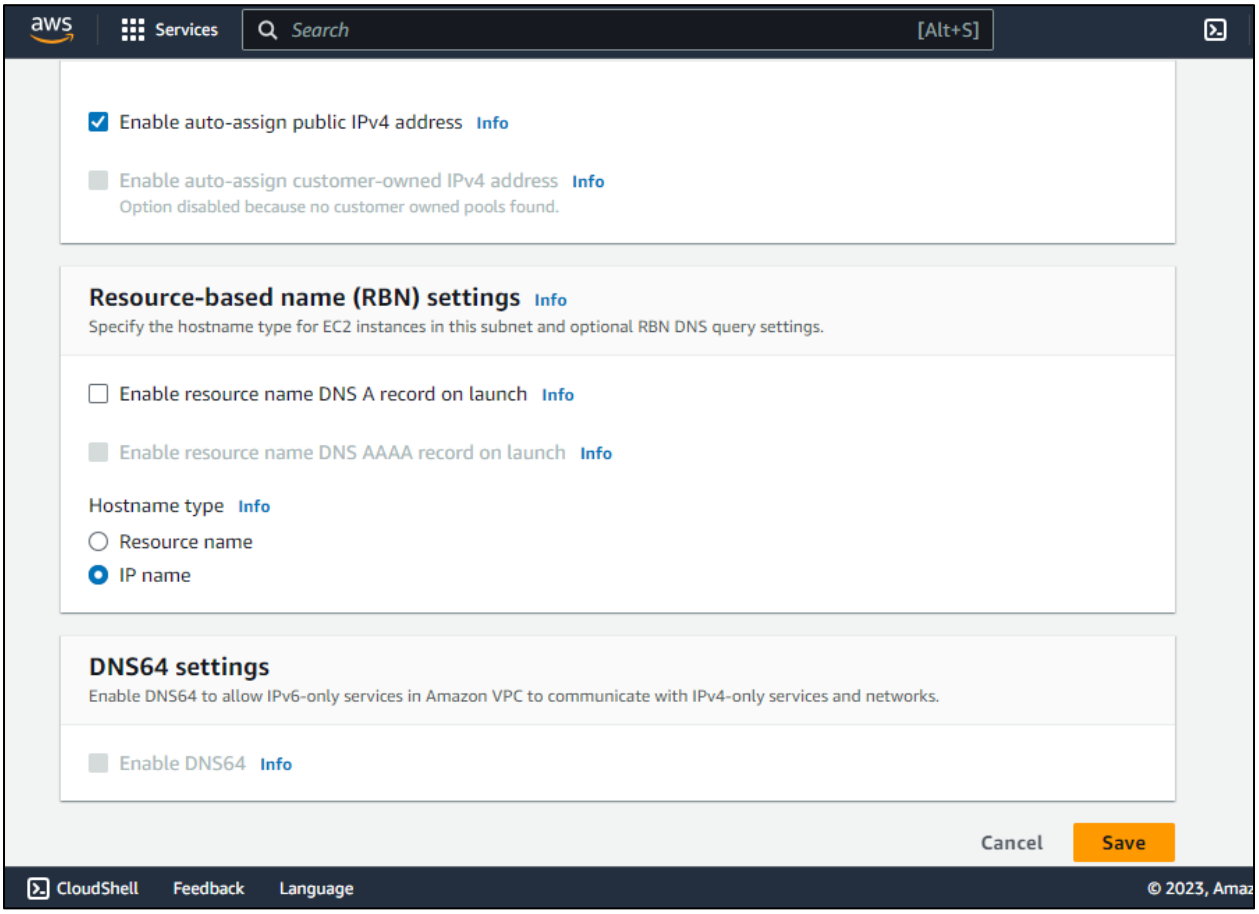
The below screen will appear after you click on **Create subnet**:



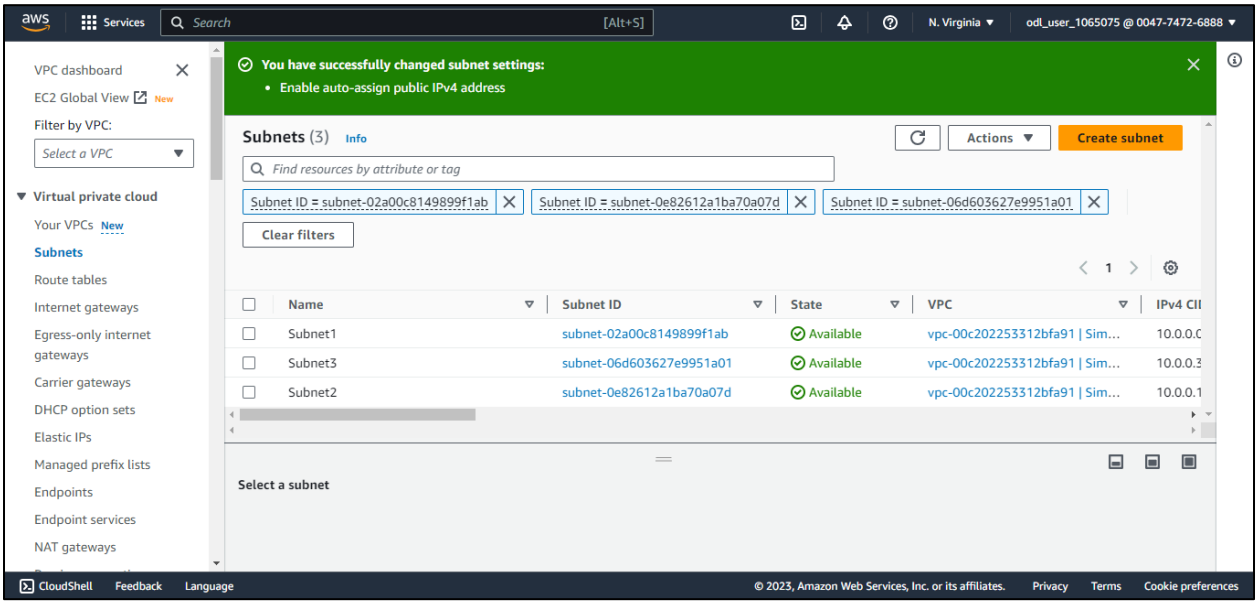
3.4 In the **Subnets** dashboards, click on **Actions** and select **Edit subnet settings**



3.5 Click on **Enable auto-assign public IPv4 address** and click on **Save**

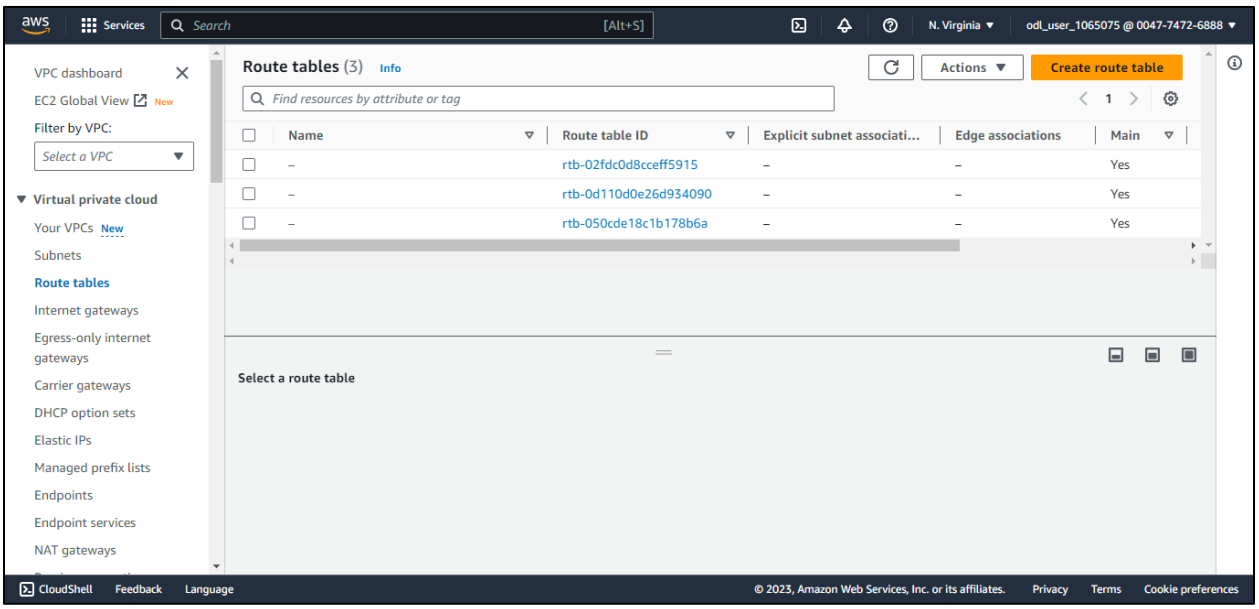


The below screen will appear after you change the subnet settings:



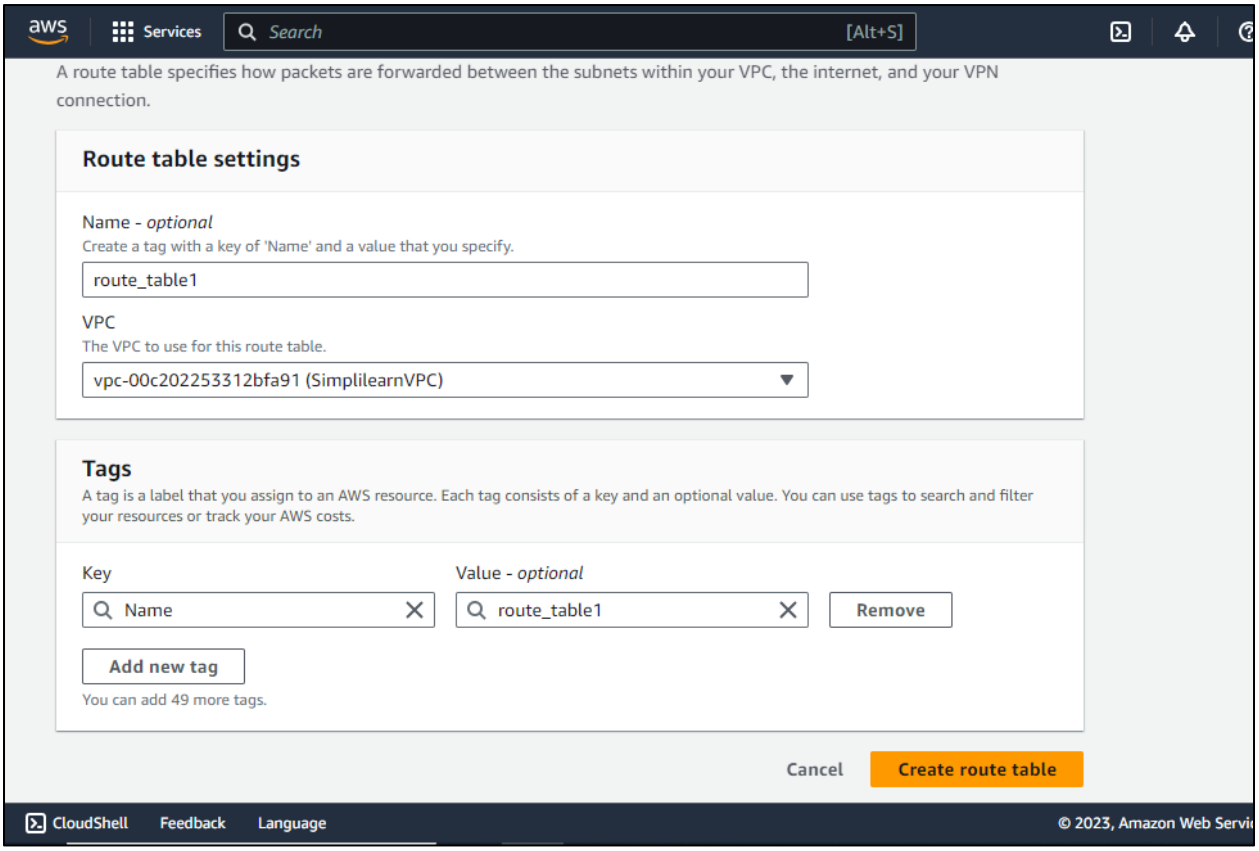
Step 4: Create a Route table and attach it to three subnets

4.1 Navigate to the **Route tables** dashboard and click on **Create route table**

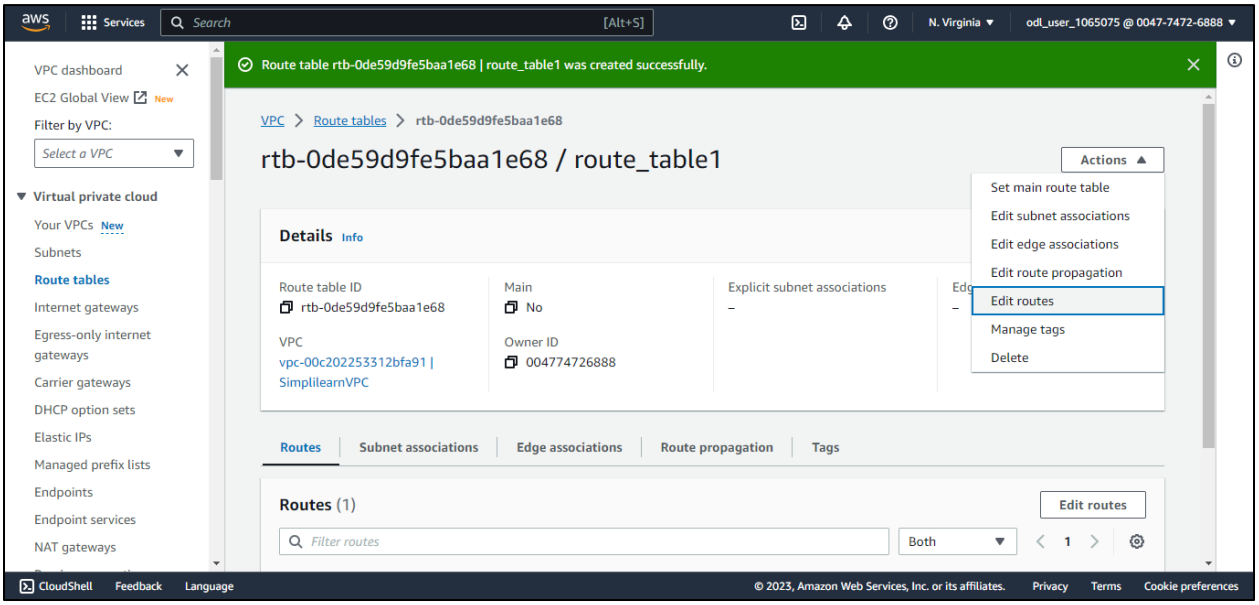


4.2 In the create route table window, perform the following:

- Enter an arbitrary name for the route table
- Add the VPC created in Step 1 in the VPC section
- Click on **Create route table**

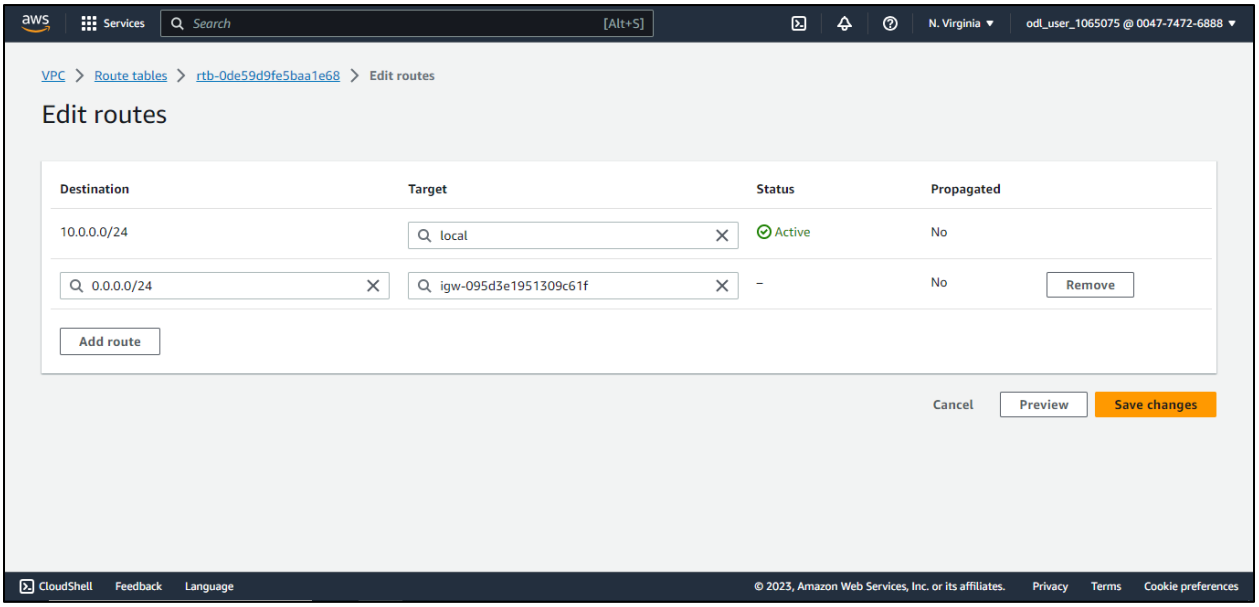


4.3 In the route table dashboard, select the route table, go to the **Actions** tab, and click on **Edit routes**

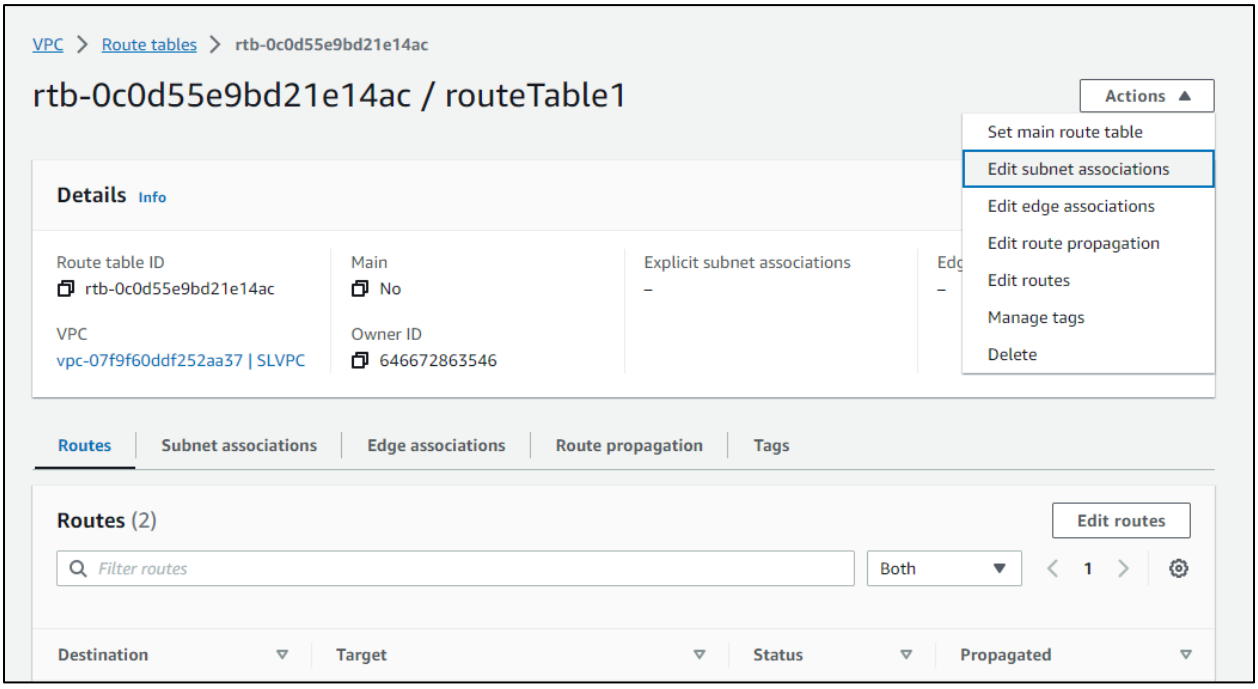


4.4 In the Edit routes window, perform the following:

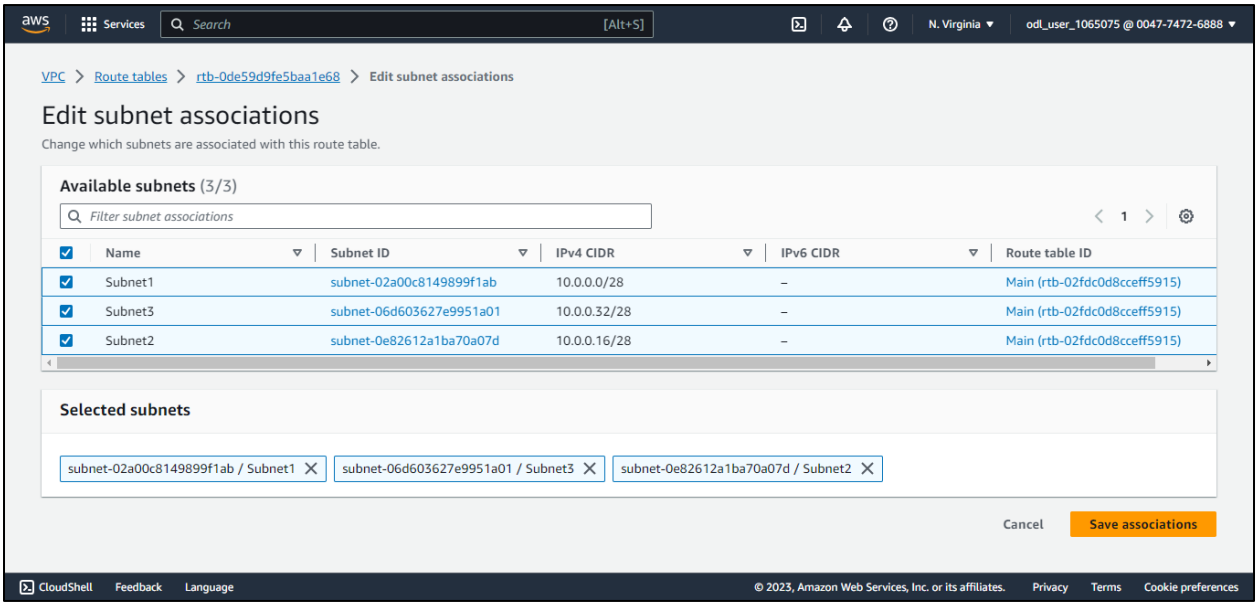
- Select **Internet gateway (igw)** from the drop-down
- Click on **Save changes**



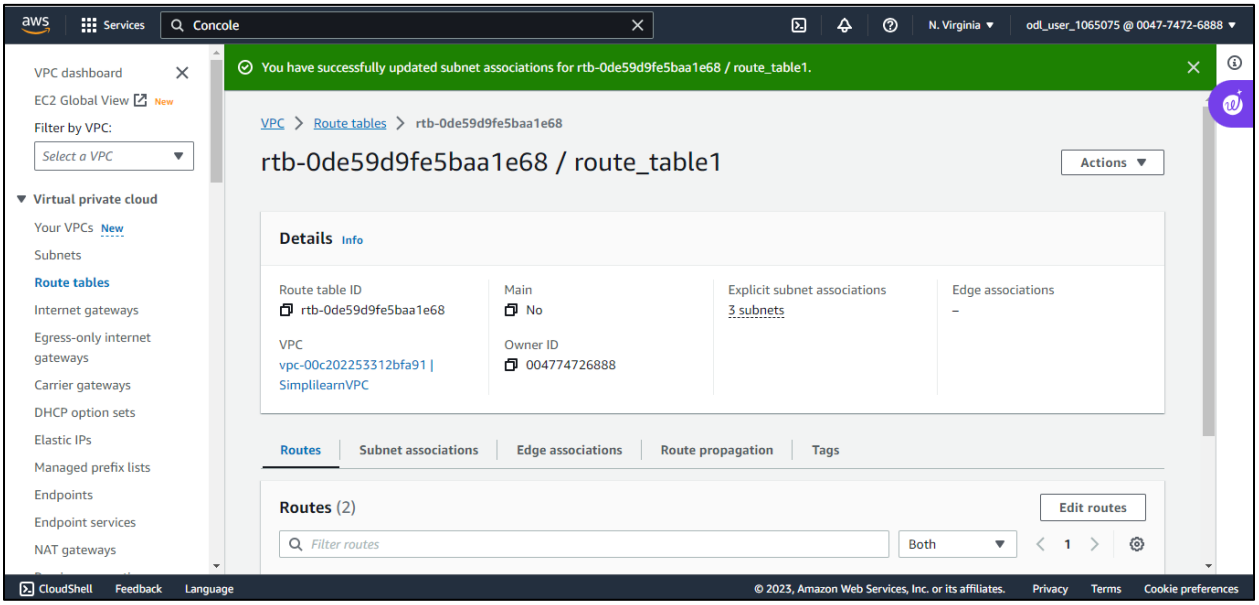
4.5 In the route table dashboard, select the route table, go to the **Actions** tab, and click on **Edit subnet associations**



4.6 Select all three subnets and click on **Save associations**

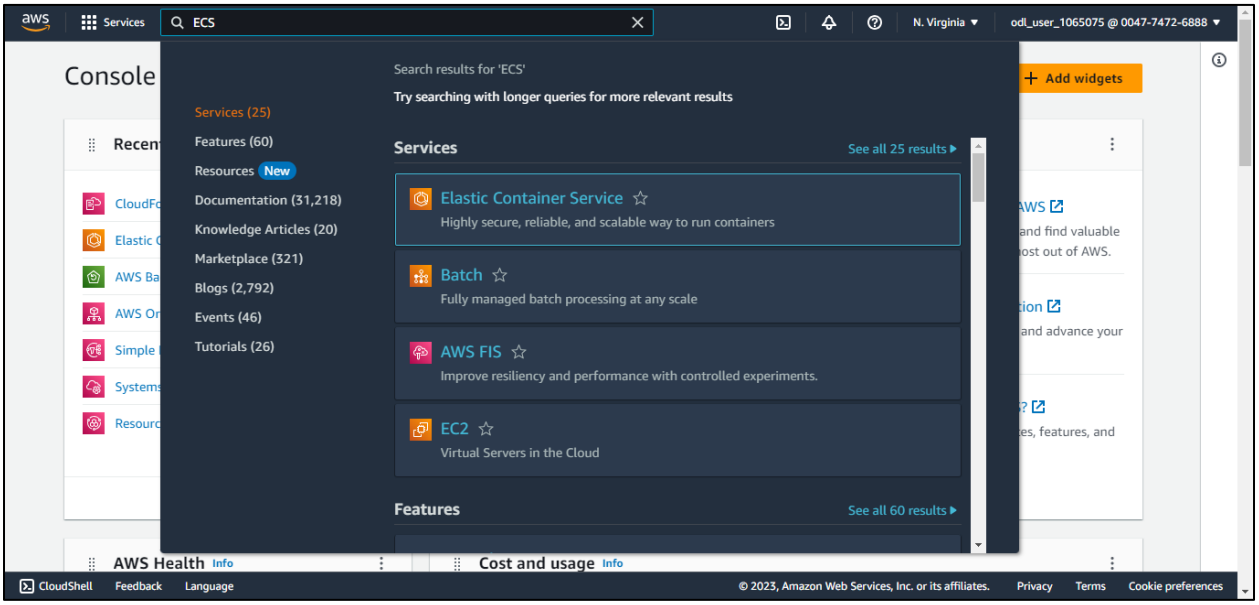


The following screen will appear:

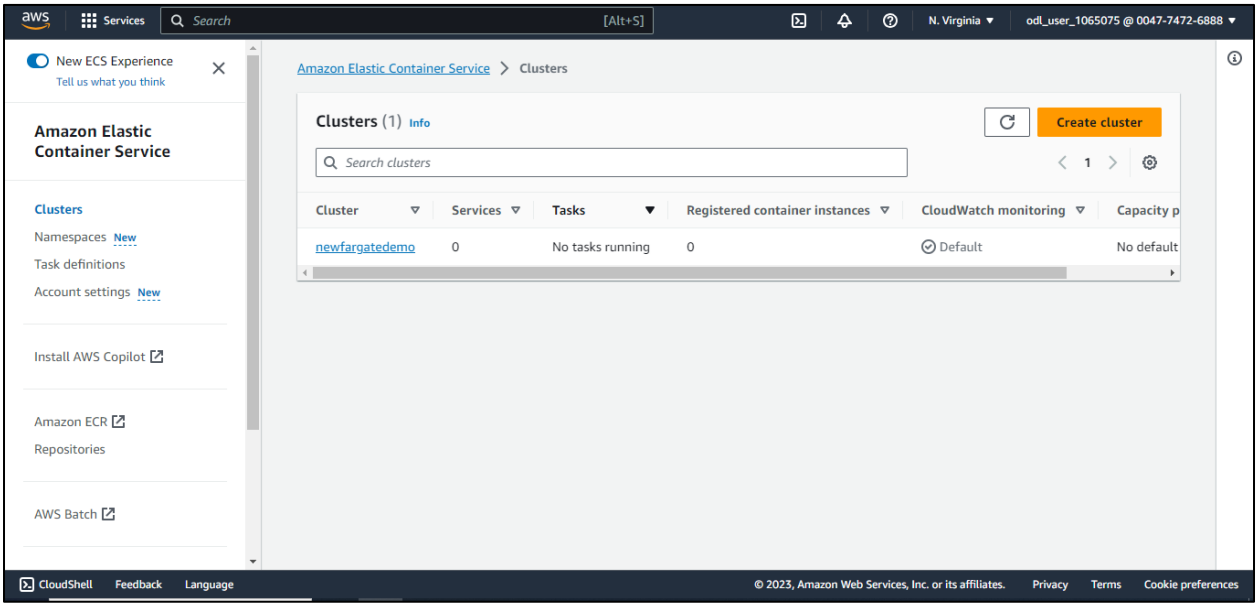


Step 7: Create a Fargate cluster

7.1 In the AWS Management Console, search for **ECS** and select **Elastic Container Service**

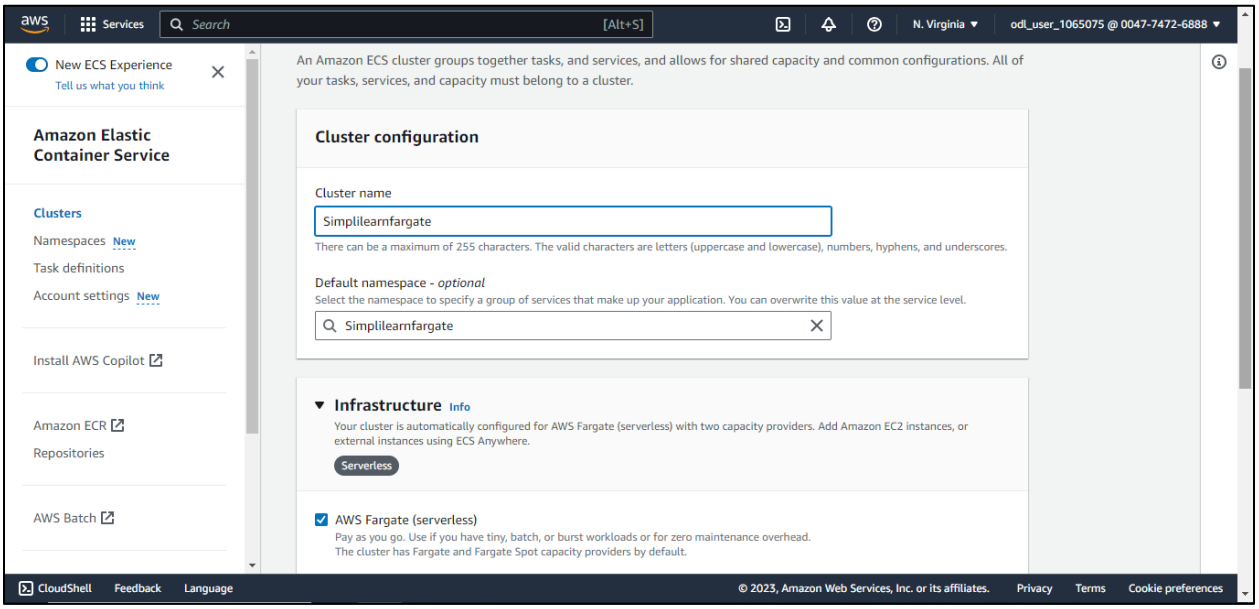


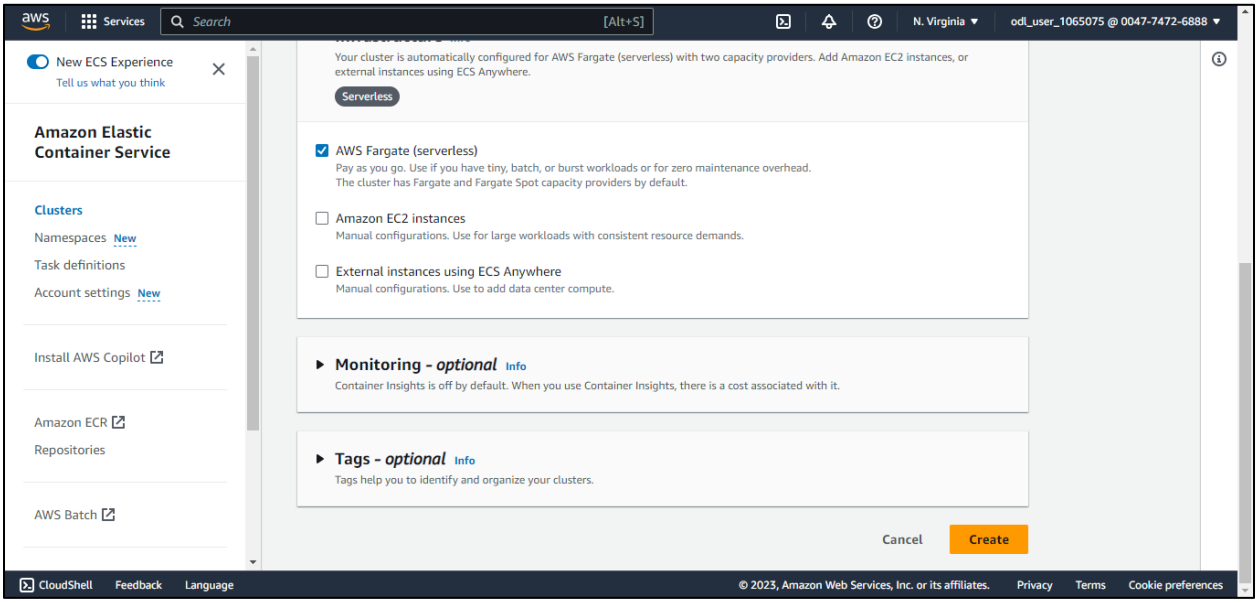
7.2 On the left panel of the ECS console, click on **Clusters** and **Create cluster**



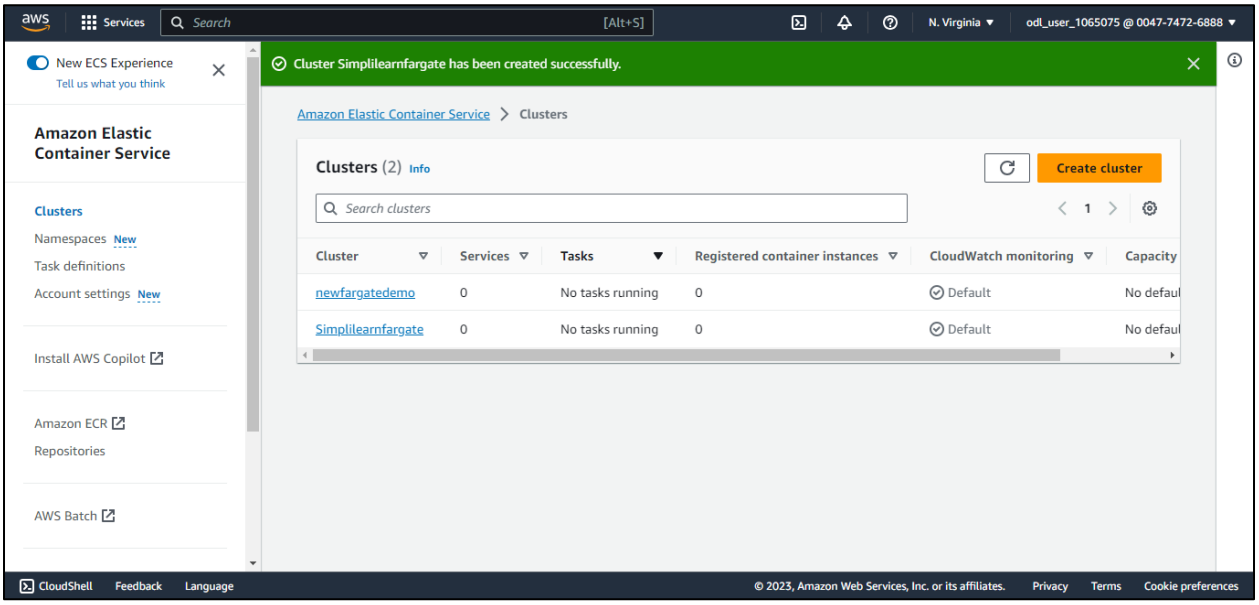
7.3 In the Cluster configuration, perform the following:

- Enter an arbitrary name for Cluster under the **Cluster name**
- In the infrastructure, specify **AWS Fargate (serverless)**
- Let the other settings stay at default settings and click on **Create**





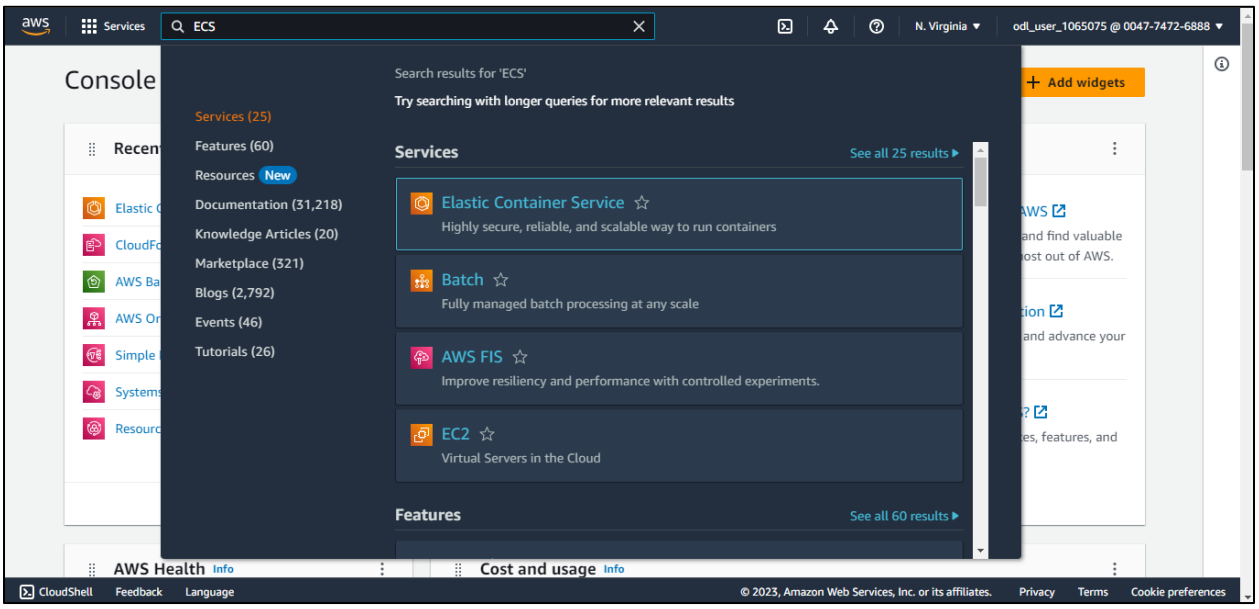
7.4 Verify the cluster creation as shown below:



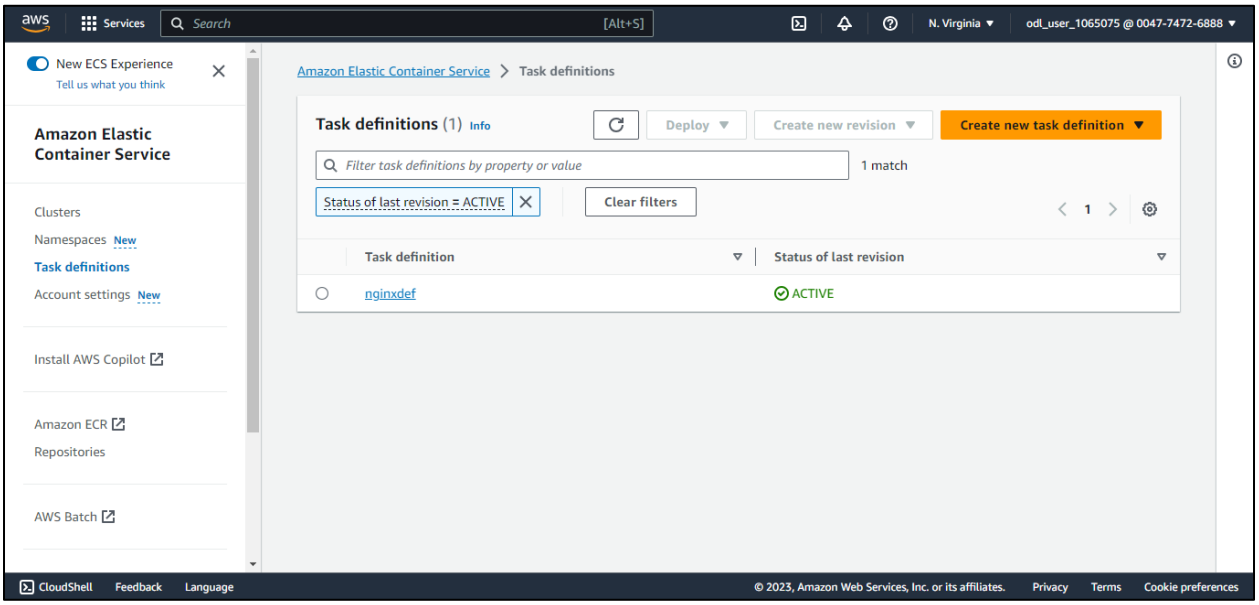
Note: Do not close the above tab. It will be necessary for reference.

Step 8: Create Task definition

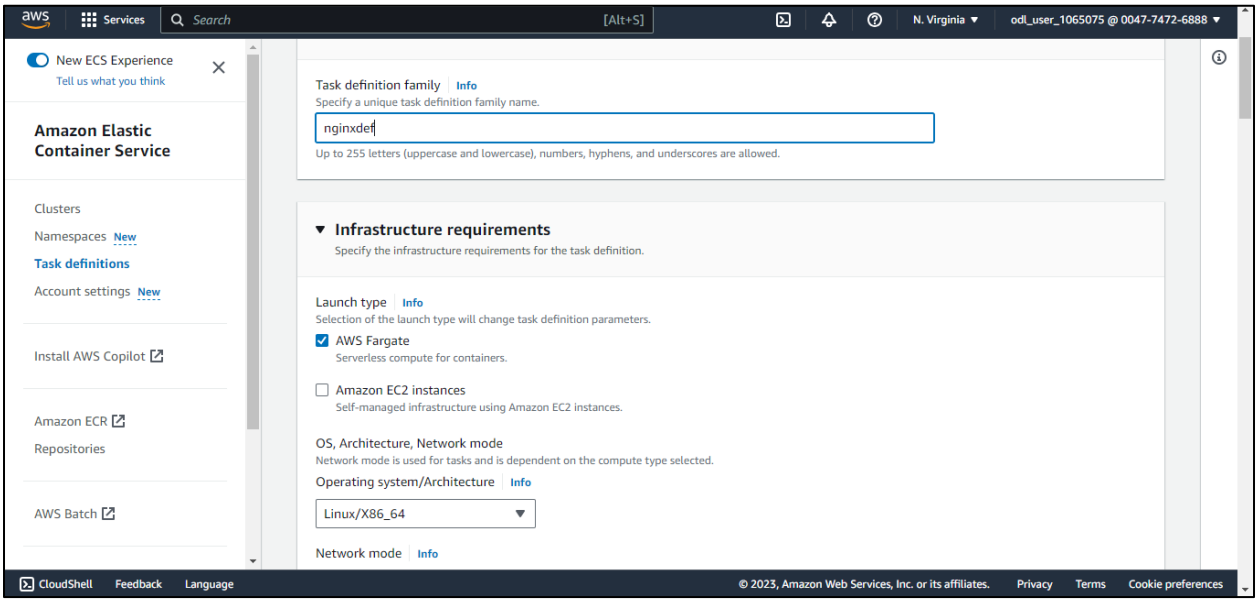
8.1 In the AWS Management Console, search for **ECS** and select **Elastic Container Service**

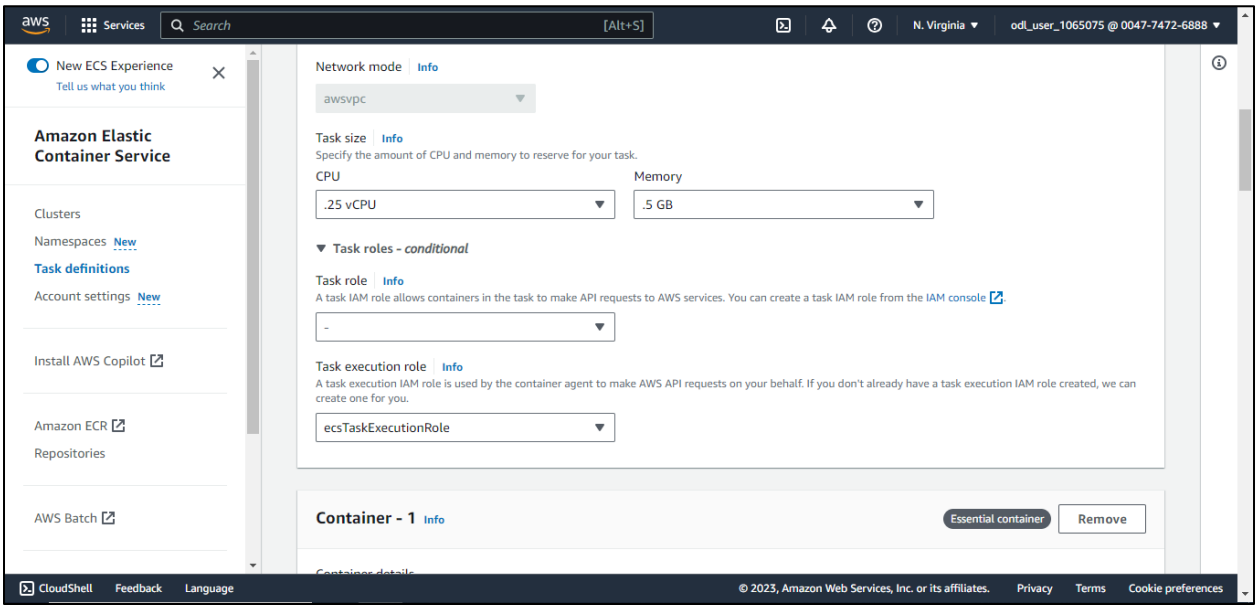


8.2 On the left panel of the ECS console, click on **Task definitions** and on **Create new task definition**



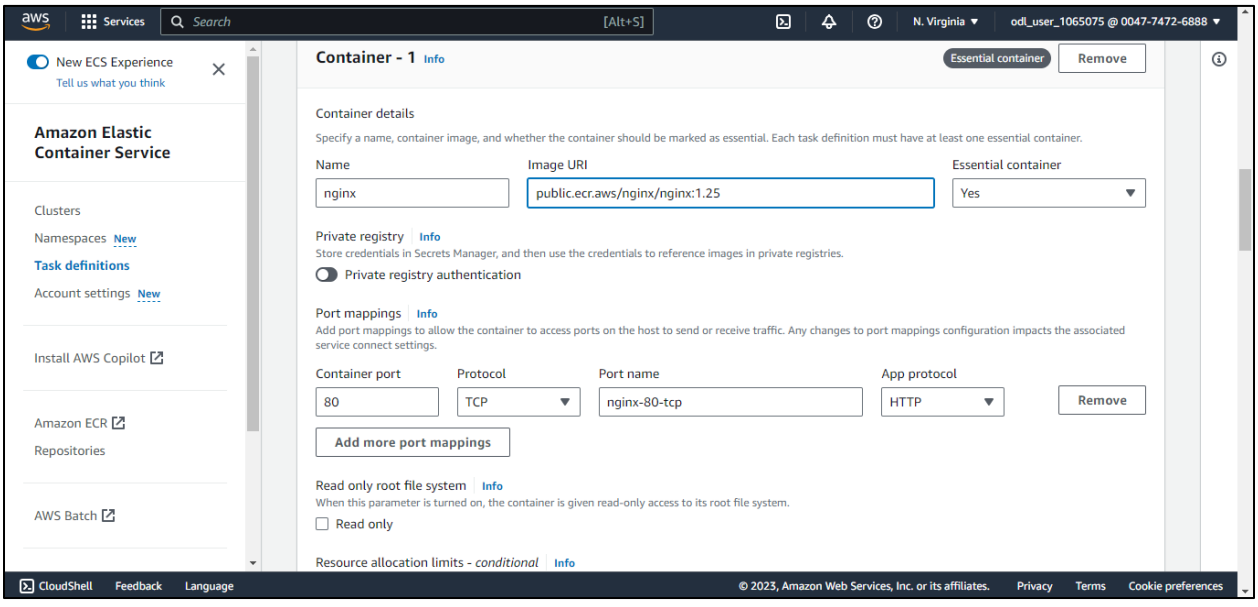
8.3 In the task definition configuration page, specify task definition family = **nginxdef,** Launch type = **AWS Fargate,** CPU = **0.25 vCPU,** and Memory = **0.5 GB**



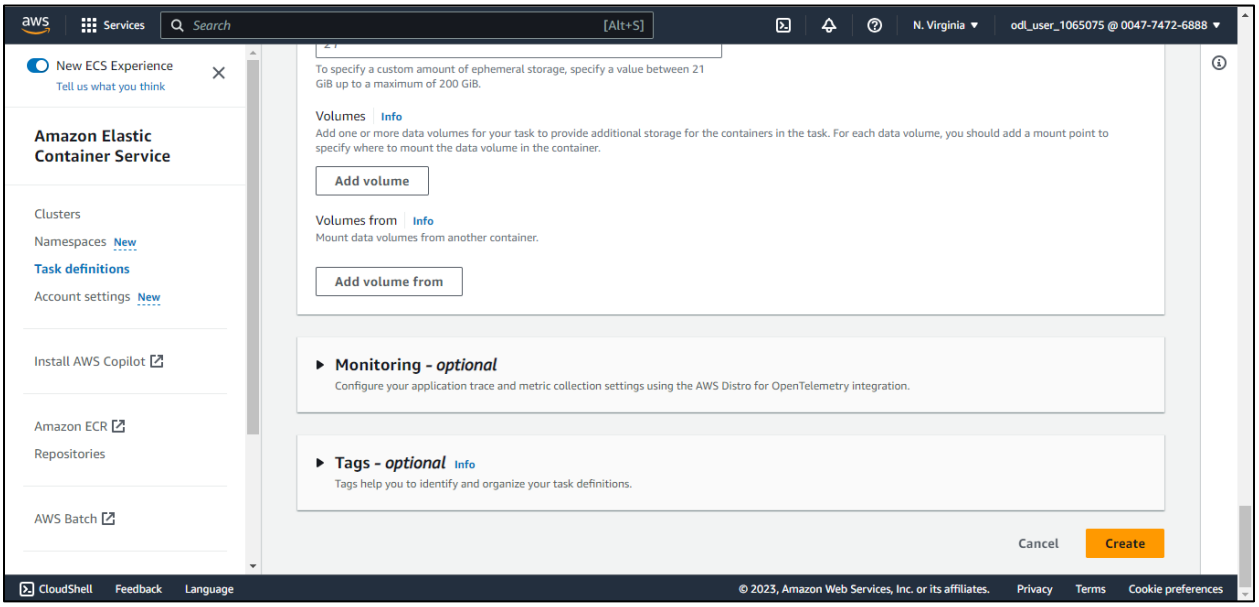


8.4 For container 1 details, enter Name = **nginx** and Image URI =

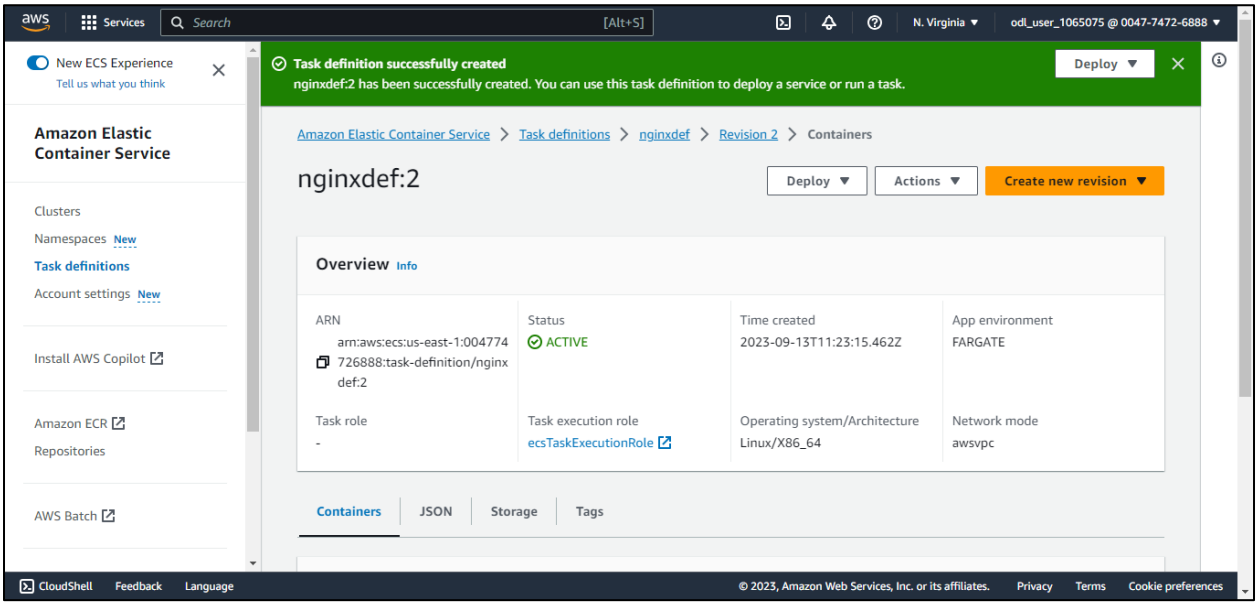
public.ecr.aws/nginx/nginx:1.25



8.5 Leave other options default and click **Create**

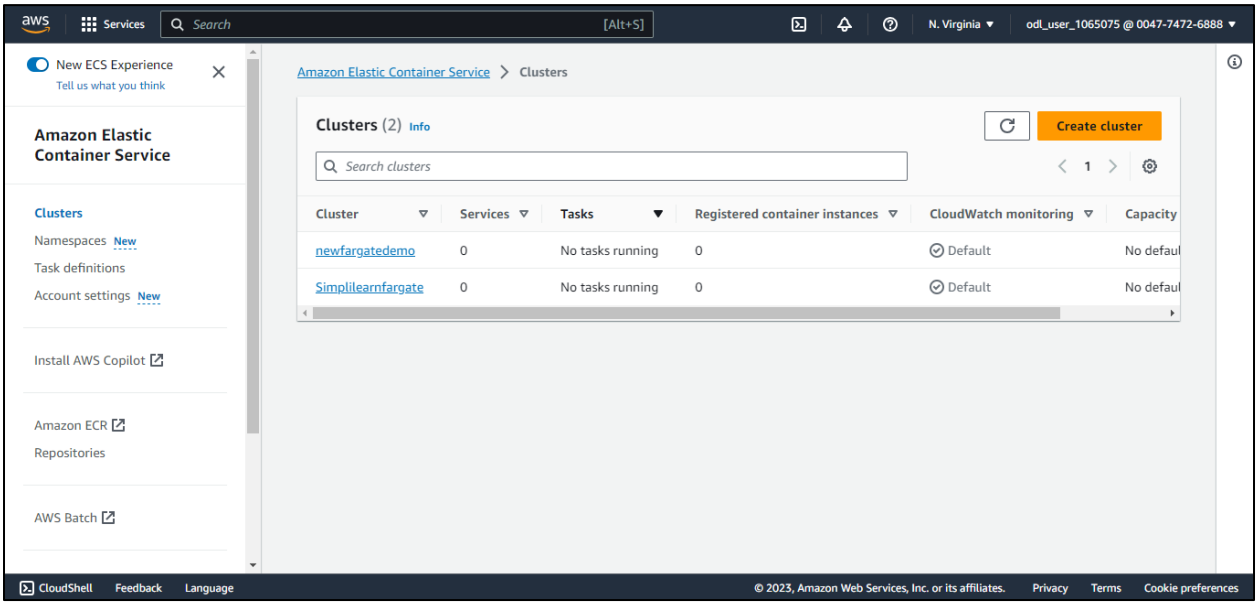


Task definition successfully created:

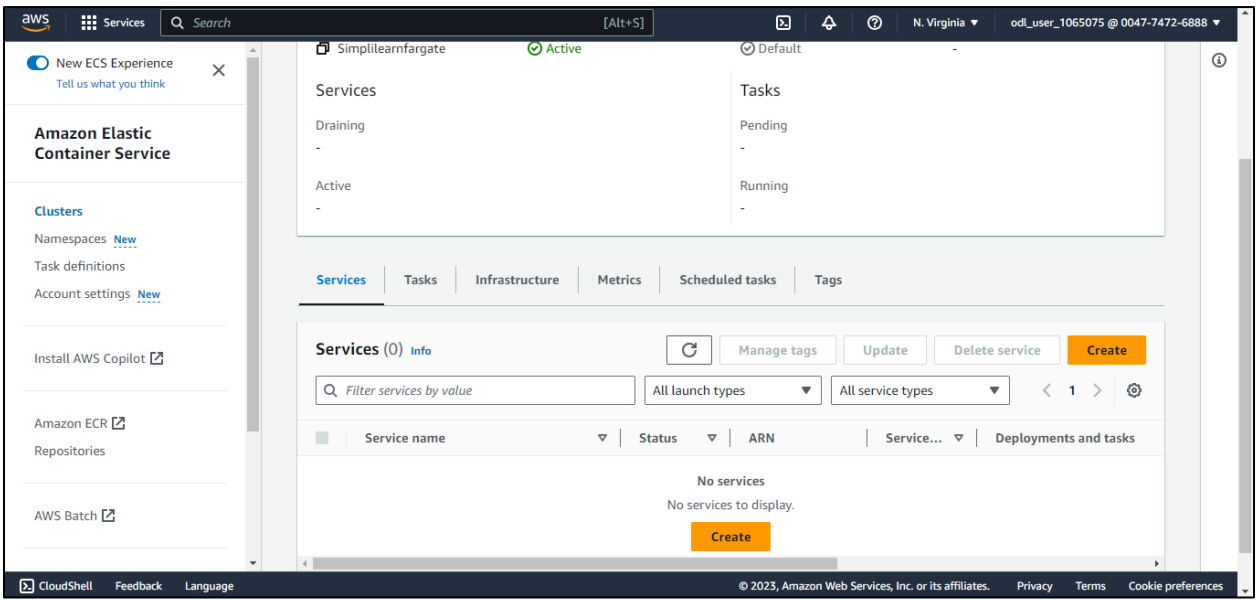


Step 9: Run its Service

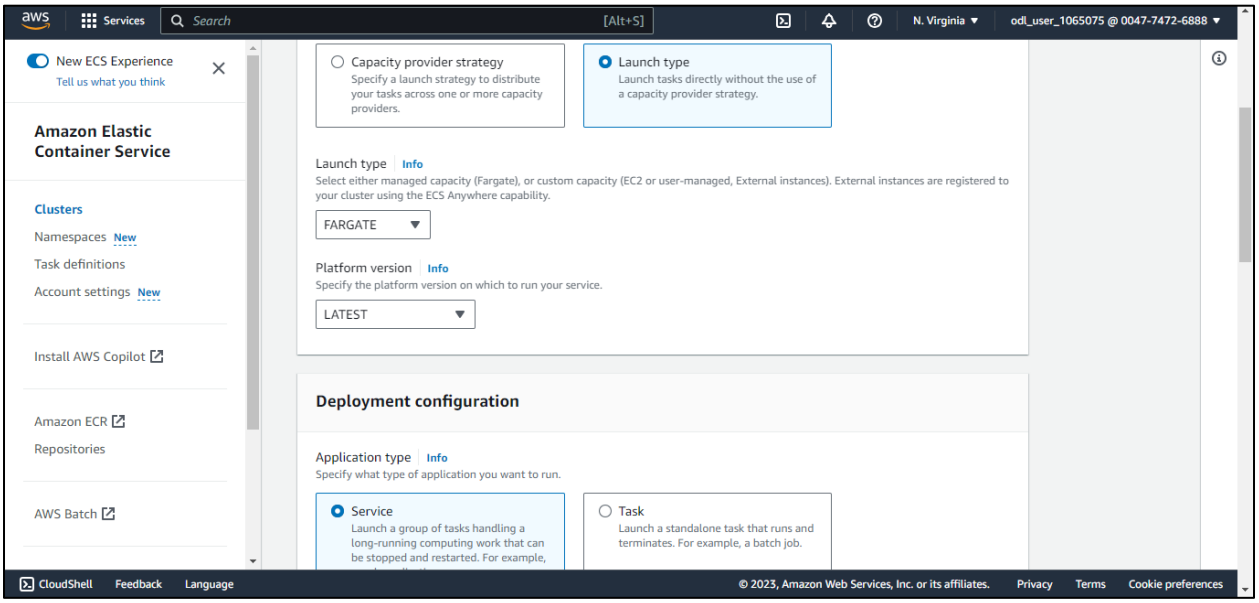
9.1 Return to the **ECS** home page, and open the newly created cluster from **Clusters**



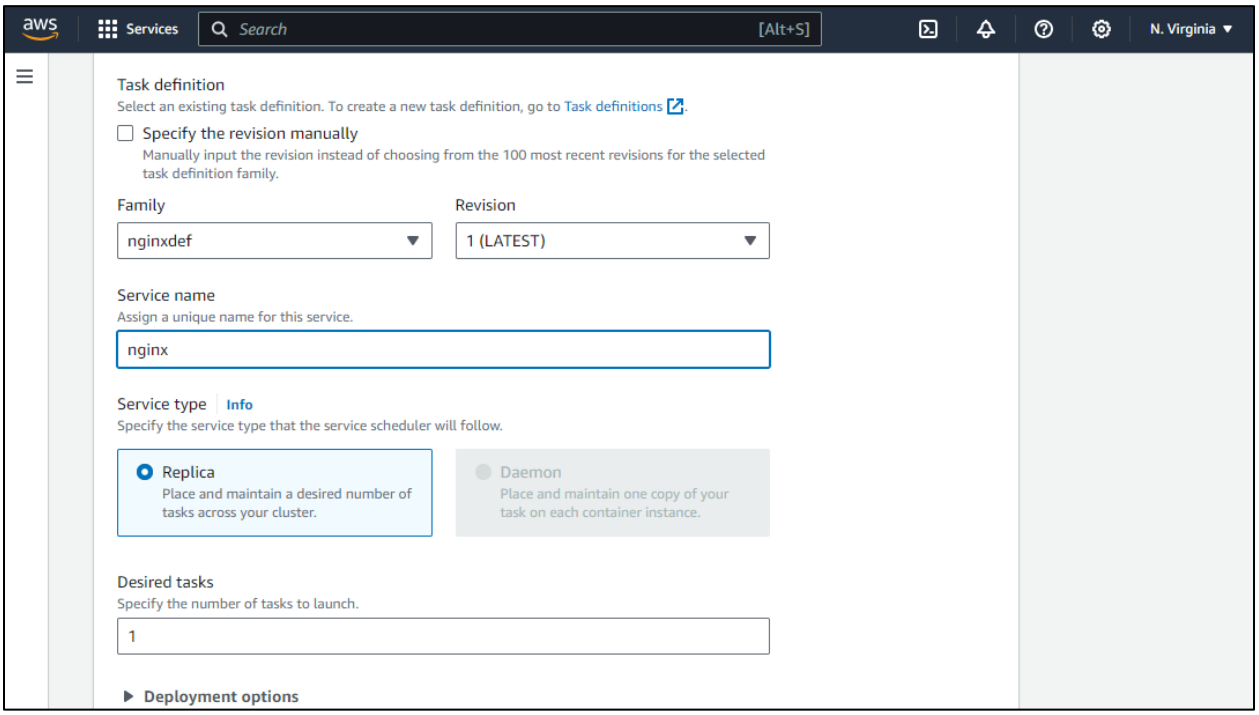
9.2 Click on **Create** under **Services**



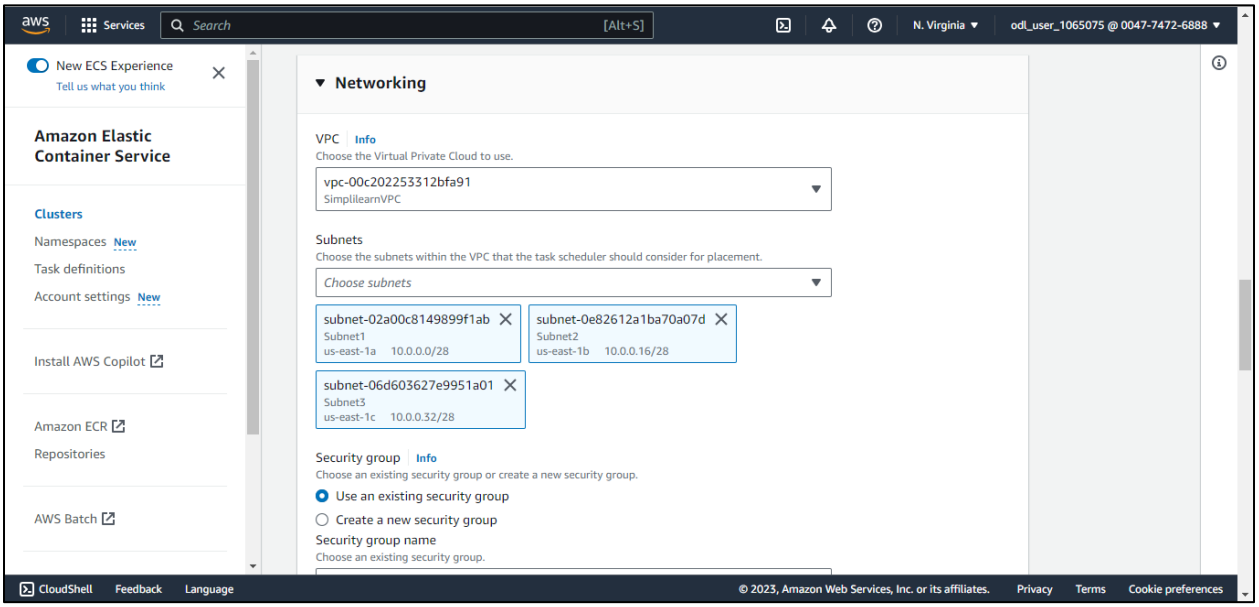
9.3 Choose **FARGATE** under Launch Type. Choose platform version as **LATEST** and application type as **Service**

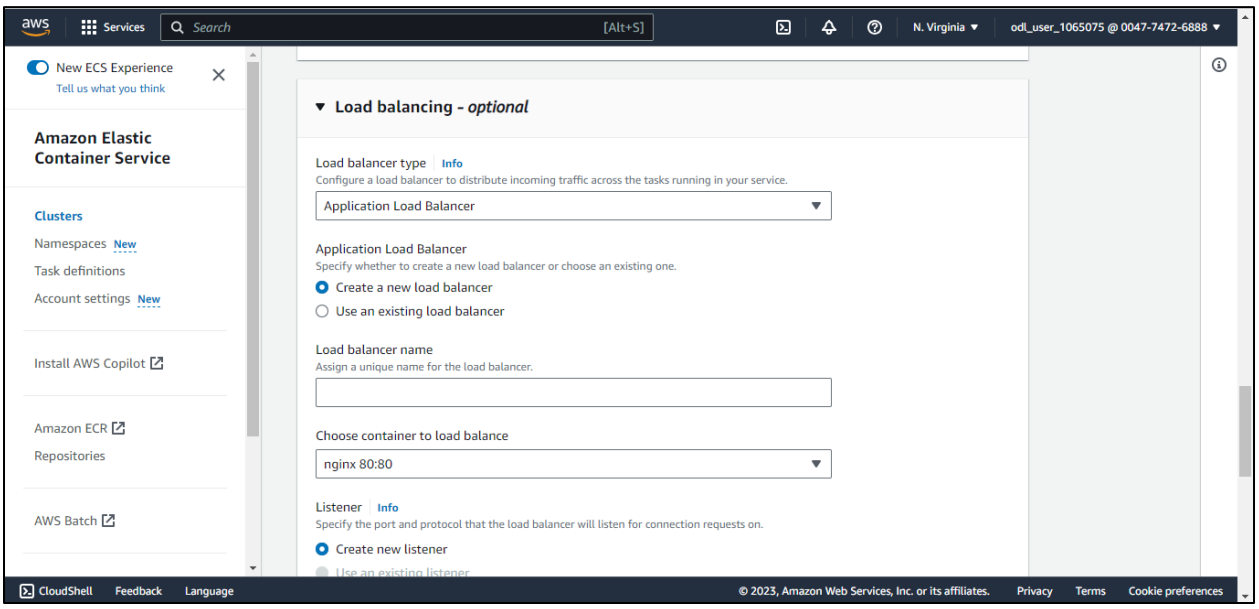


9.4 Choose family = **nginxdef** (created earlier), revision = **1 (LATEST),** service name = **nginx**, and Replica = **1**

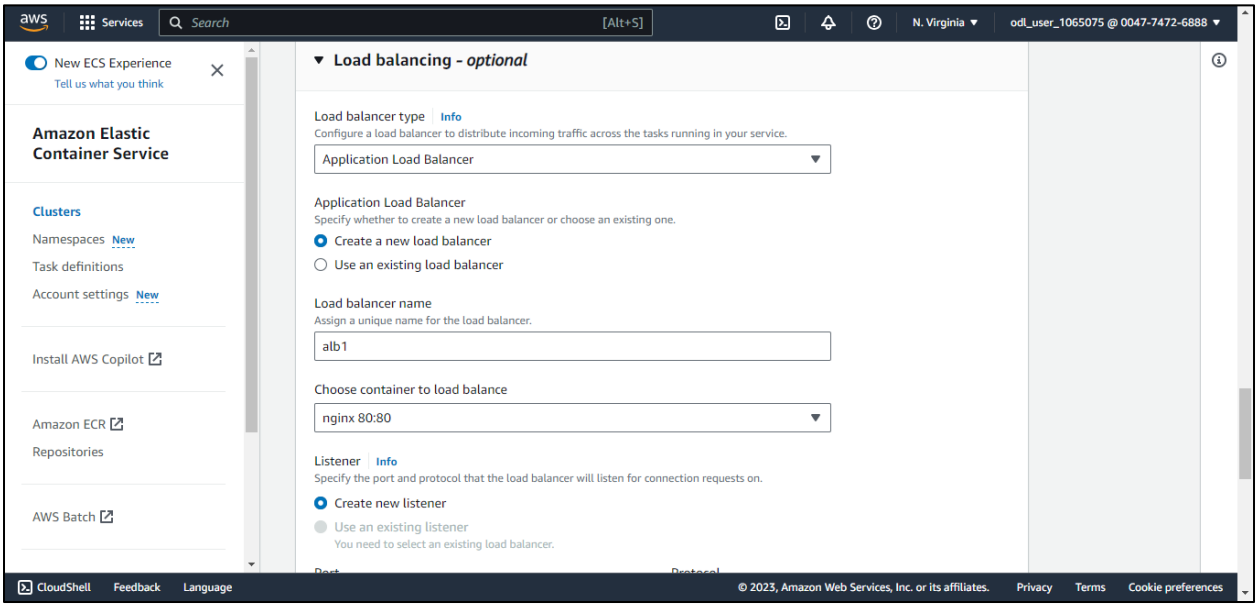


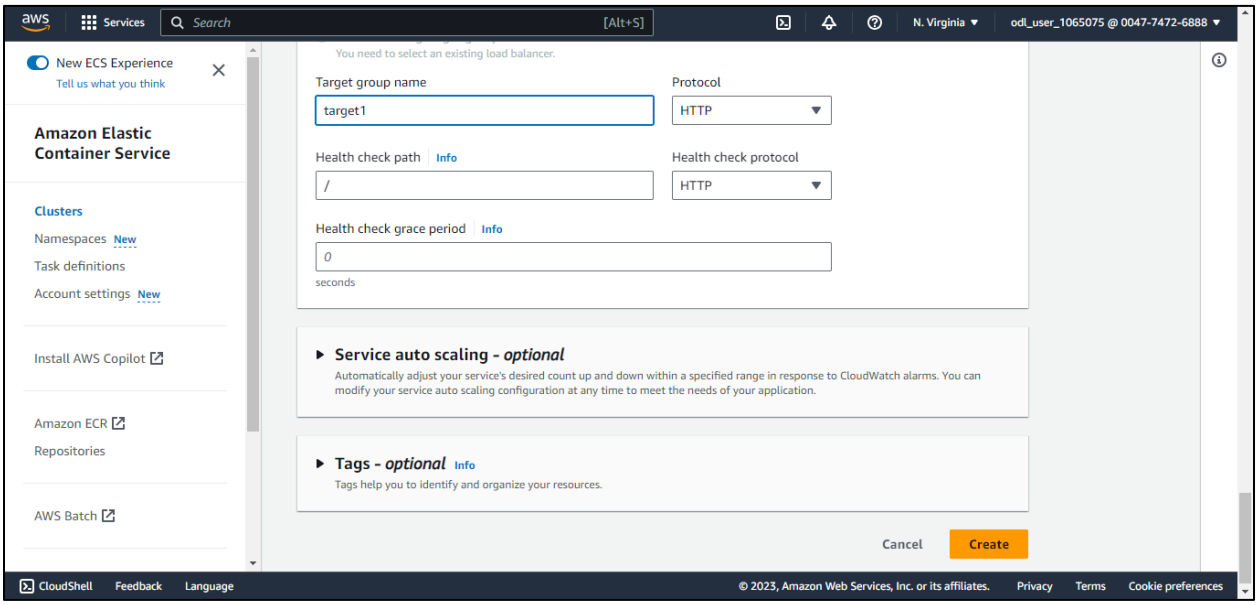
9.5 In the **Networking** section, select the previously created VPC. For Load balancing, choose **Application Load Balancer** from the drop-down options.



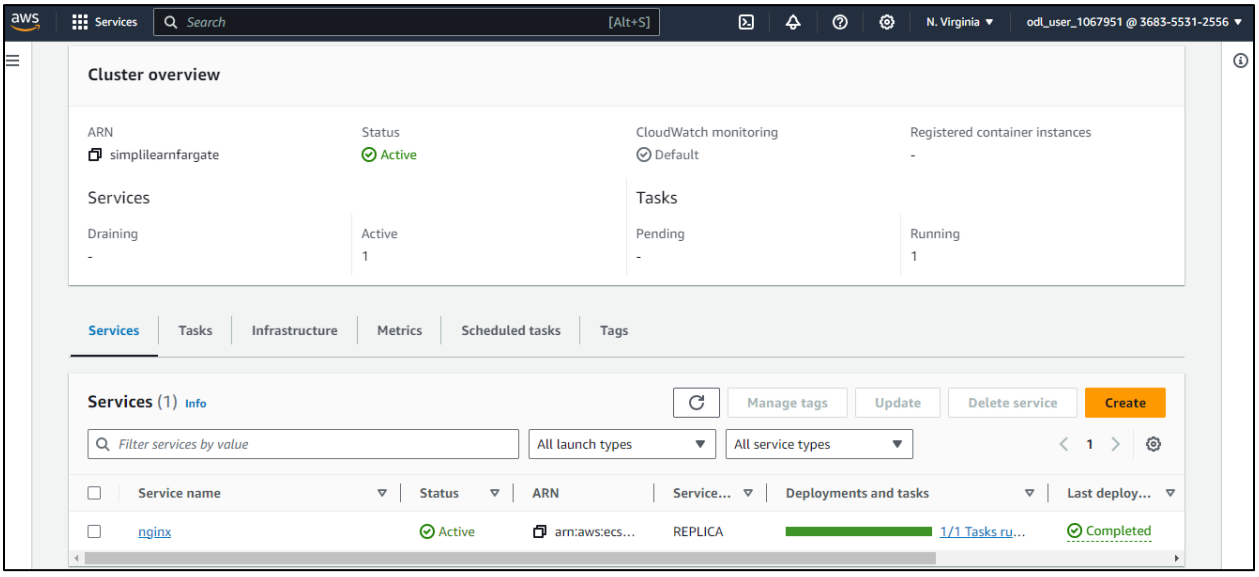


9.6 Enter Load balancer name = **alb1,** choose container to load balance = **nginx 80:80**, Listener **= Create new listener**, port = **80,** target group = **create new target group**, and target group name = **target1**

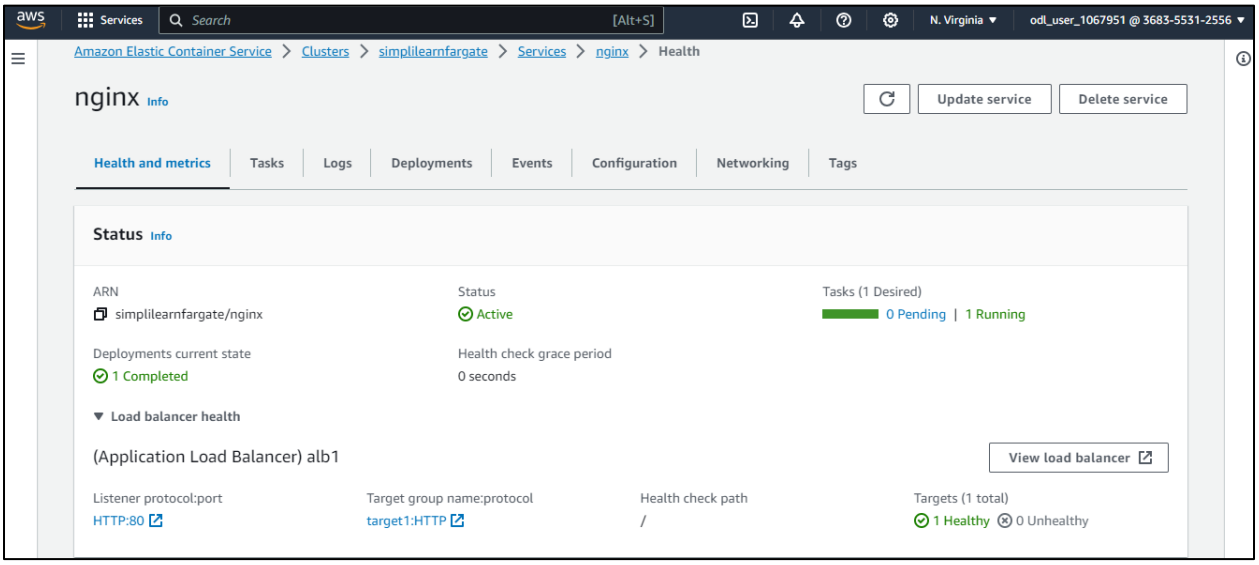




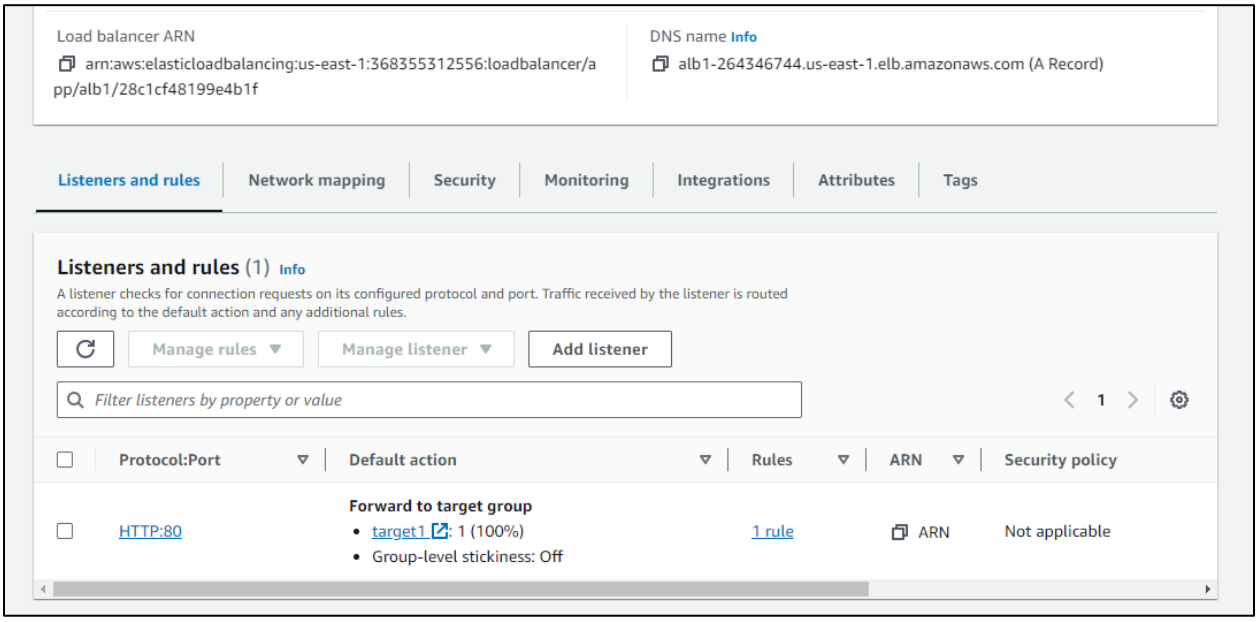
9.7 Leaving other options default, click **Create.** Wait until service creation has been completed and 1/1 tasks are shown as active and running.



9.8 Once the service is running successfully, click on **nginx** in the above screen and view the service details below. Click on **View load balancer** to get the load balancer URL, which will help in accessing the application running on the ECS cluster with the container:

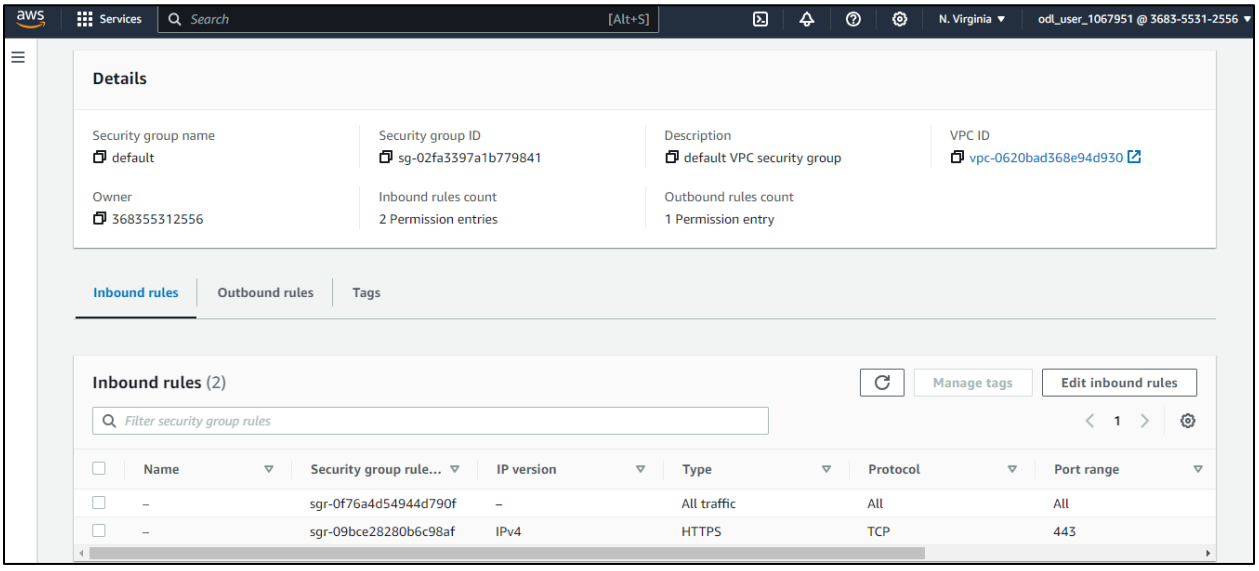


9.9 Copy the DNS name and open it in a new browser



9.10 Open the URL in a new browser. You will find the nginx page loading as shown below:

Note: In case, the web page doesn't load, go to **Load balancer > Security > Open security group**, and select the security group being used to ensure the port 80 inbound rule is accessible from anywhere as shown below:



Hence, you have successfully mapped the port with ALB.