Lesson 10 Demo 04

Integrating Dynamic Port with Application Load Balancer

Objective: To integrate Dynamic Port Mapping with an Application Load Balancer for efficient distribution of traffic to multiple instances of a containerized application

Tools required: AWS account

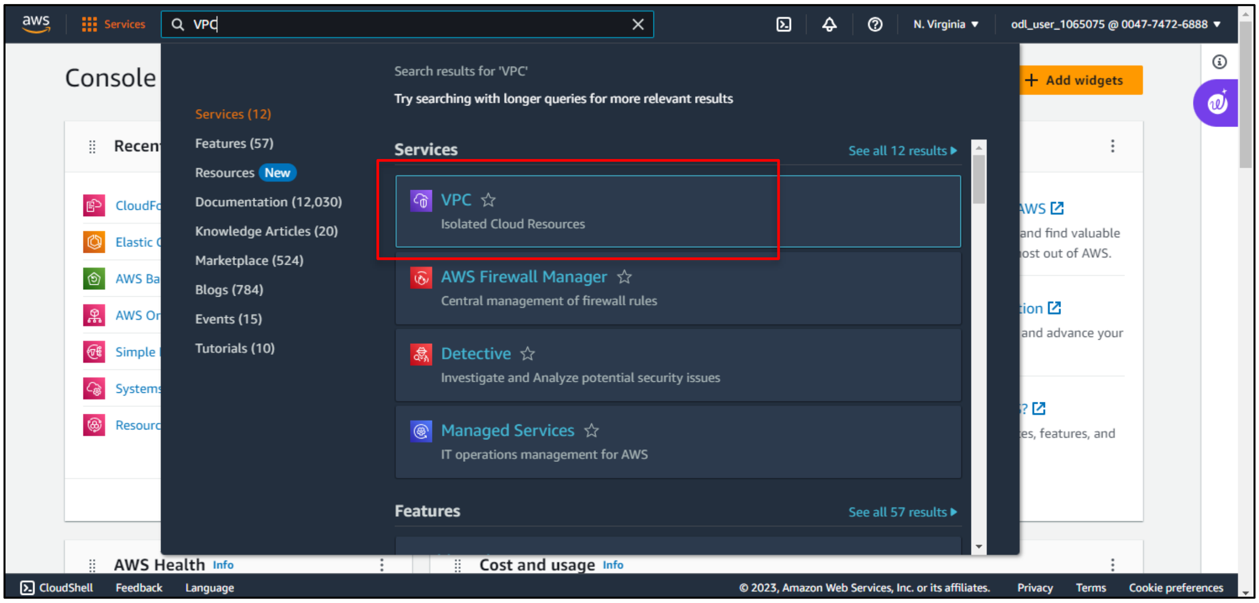
Prerequisites: NA

Steps to be followed:

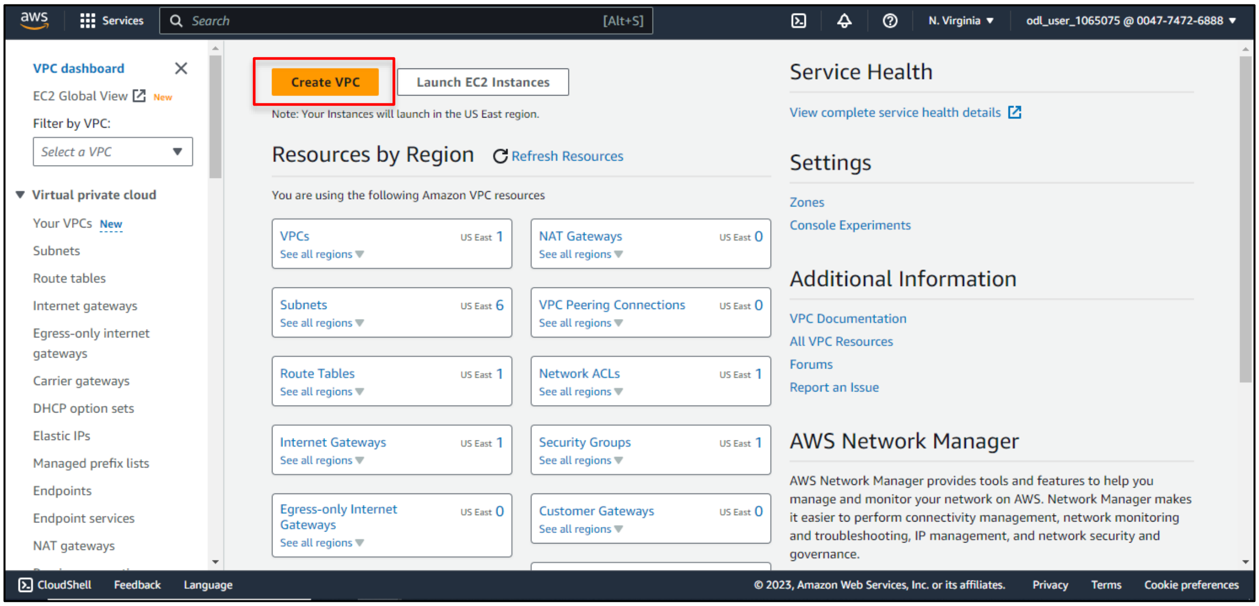
1. Create a custom VPC and enable the DNS hostname
2. Create an Internet Gateway
3. Create three subnets
4. Create a route table and attach it to three subnets
5. Create a cluster
6. Create a task definition
7. Run the service on the cluster

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

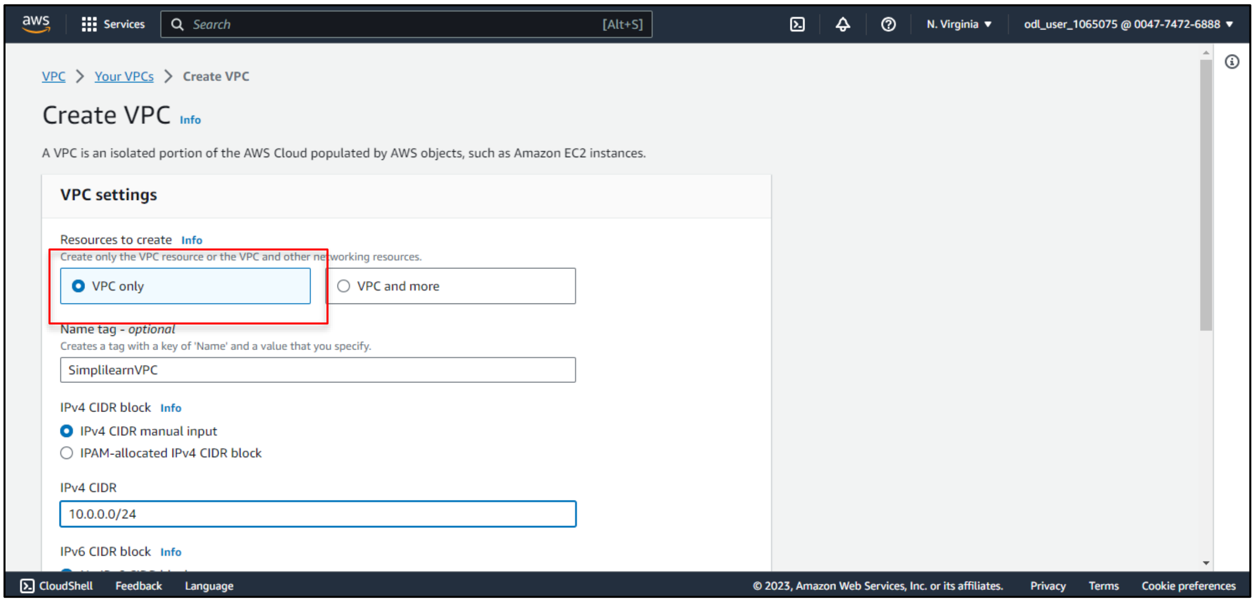
1.1 Navigate to the AWS Management Console, search for and click **VPC**



1.2 Click on **Create VPC** in the VPC dashboard


aws
Services
Search
[Alt+S]
N. Virginia
odl_user_1065075 @ 0047-7472-6888
VPC dashboard
EC2 Global View New
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs New
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Create VPC
Launch EC2 Instances
Note: Your Instances will launch in the US East region.
Resources by Region
Refresh Resources
You are using the following Amazon VPC resources
VPCs US East 1 See all regions
NAT Gateways US East 0 See all regions
Subnets US East 6 See all regions
VPC Peering Connections US East 0 See all regions
Route Tables US East 1 See all regions
Network ACLs US East 1 See all regions
Internet Gateways US East 1 See all regions
Security Groups US East 1 See all regions
Egress-only Internet Gateways US East 0 See all regions
Customer Gateways US East 0 See all regions
Service Health
View complete service health details
Settings
Zones
Console Experiments
Additional Information
VPC Documentation
All VPC Resources
Forums
Report an Issue
AWS Network Manager
AWS Network Manager provides tools and features to help you manage and monitor your network on AWS. Network Manager makes it easier to perform connectivity management, network monitoring and troubleshooting, IP management, and network security and governance.
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

1.3 Select **VPC only** under the **Resource to create** section



aws
Services
Search
[Alt+S]
N. Virginia ▼
odl_user_1065075 @ 0047-7472-6888 ▼
VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.
● VPC only ○ VPC and more

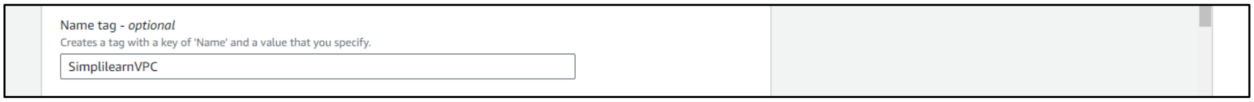
Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
SimplilearnVPC

IPv4 CIDR block Info
● IPv4 CIDR manual input
○ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/24

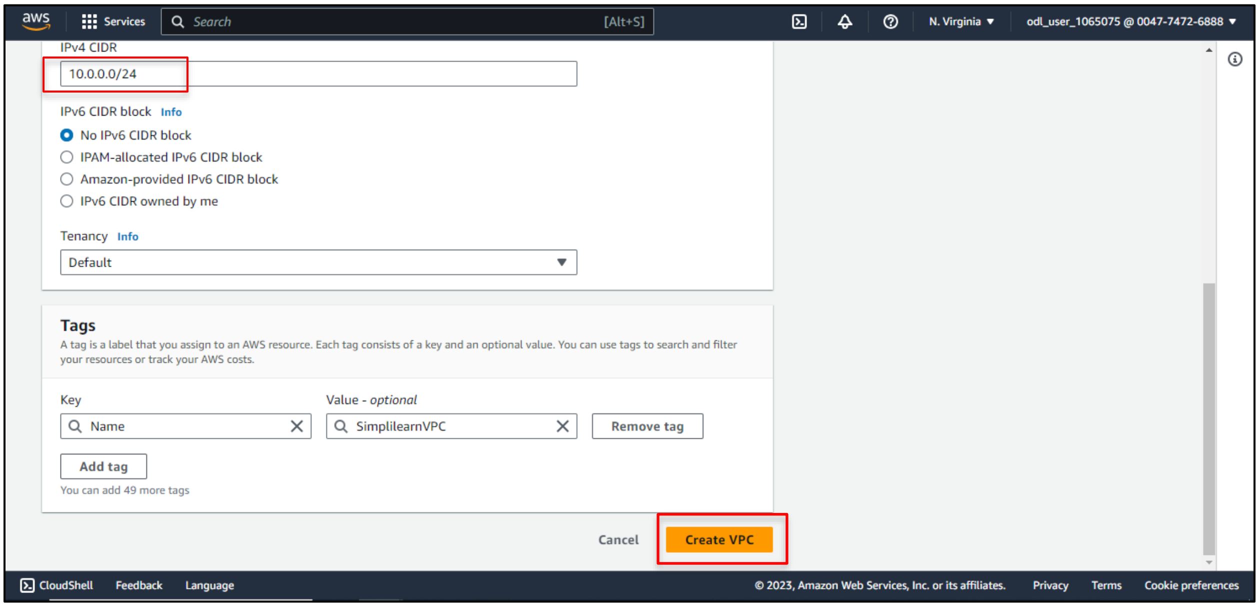
IPv6 CIDR block Info

1.4 Enter an arbitrary name for the VPC under the **Name tag** section



Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
SimplilearnVPC

1.5 Enter **10.0.0.0/24** in the **IPv4 CDR** and click on **Create VPC**



IPv4 CIDR
10.0.0.0/24

IPv6 CIDR block Info

● No IPv6 CIDR block
○ IPAM-allocated IPv6 CIDR block
○ Amazon-provided IPv6 CIDR block
○ IPv6 CIDR owned by me

Tenancy Info
Default ▼

Tags

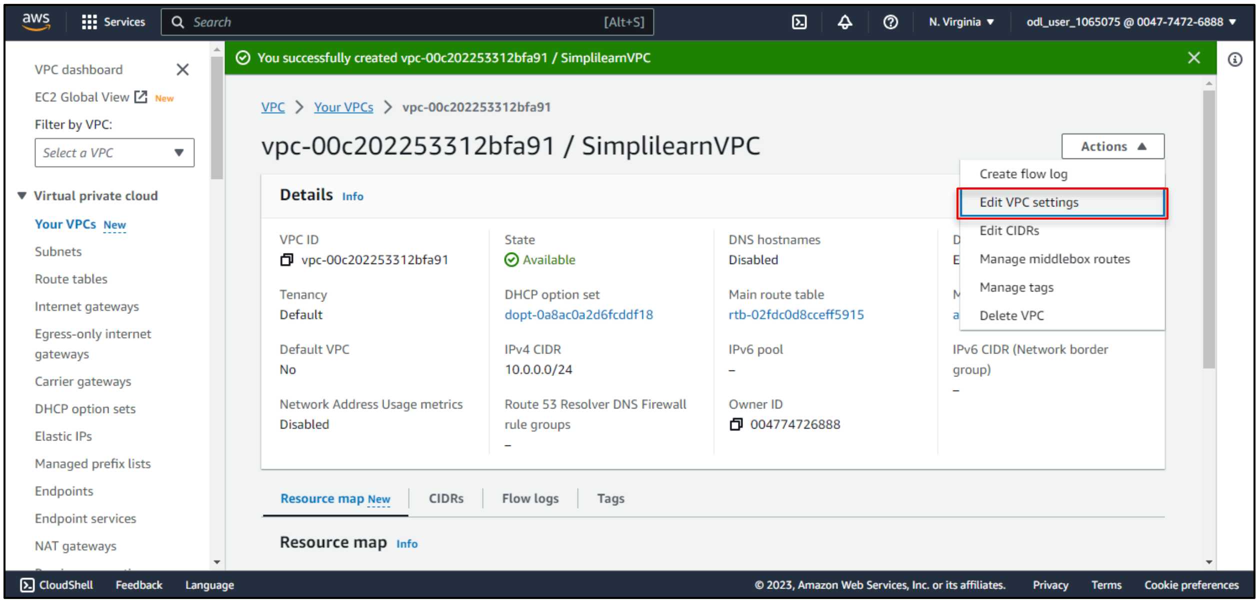
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
Name	SimplilearnVPC	Remove tag

Add tag

You can add 49 more tags

1.6 Click on the **Actions** dropdown menu and select **Edit VPC Settings** in the VPC dashboard



You successfully created vpc-00c202253312bfa91 / SimplilearnVPC

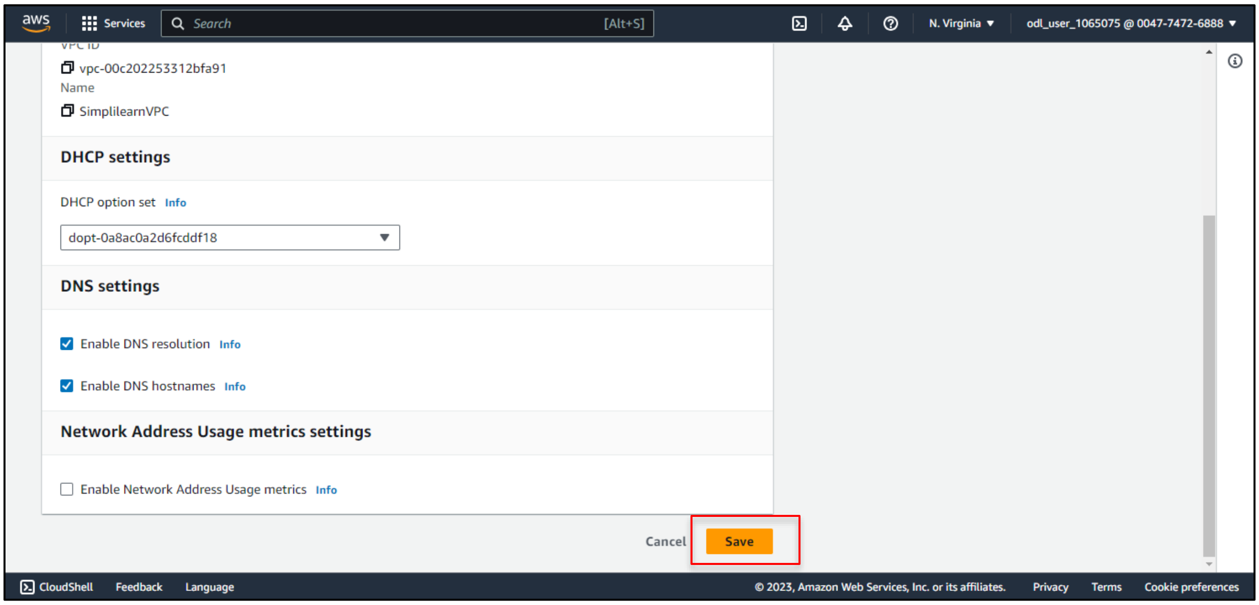
vpc-00c202253312bfa91 / SimplilearnVPC

Details Info

VPC ID vpc-00c202253312bfa91	State Available	DNS hostnames Disabled	D E
Tenancy Default	DHCP option set dopt-0a8ac0a2d6fcddf18	Main route table rtb-02fdc0d8cceff5915	M a
Default VPC No	IPv4 CIDR 10.0.0.0/24	IPv6 pool –	IPv6 CIDR (Network border group) –
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups –	Owner ID 004774726888	

Resource map Info

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



VPC ID
vpc-00c202253312bfa91
Name
SimplilearnVPC

DHCP settings

DHCP option set Info
dopt-0a8ac0a2d6fcddf18

DNS settings

☑ Enable DNS resolution Info

☑ Enable DNS hostnames Info

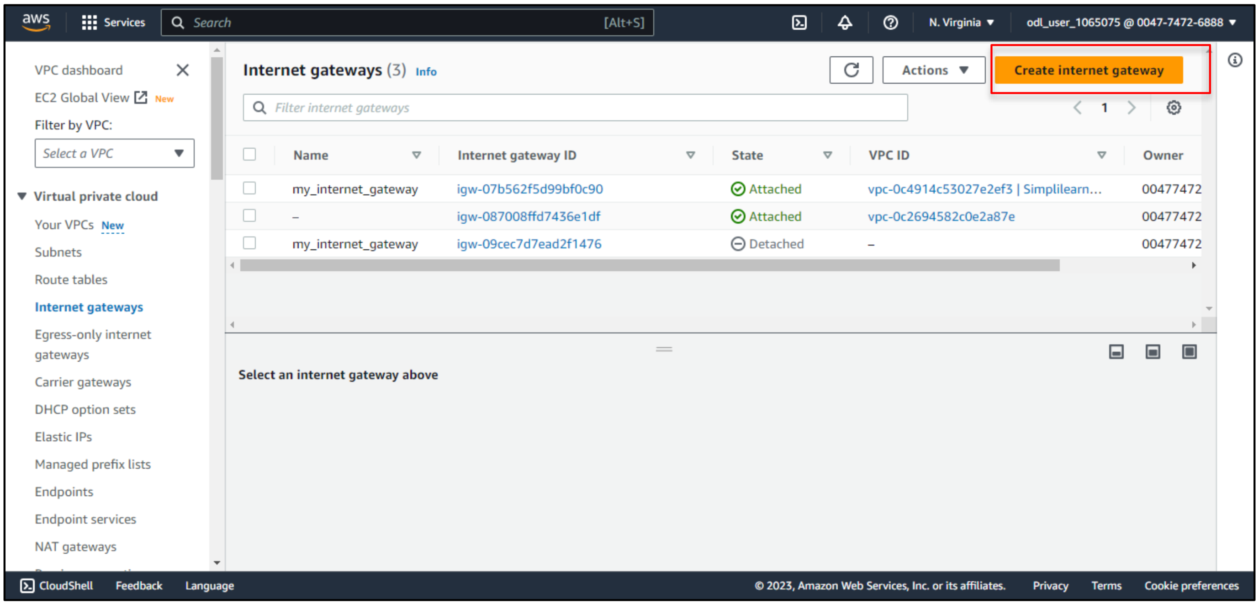
Network Address Usage metrics settings

☐ Enable Network Address Usage metrics Info

Cancel Save

Step 2: Create an Internet gateway

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

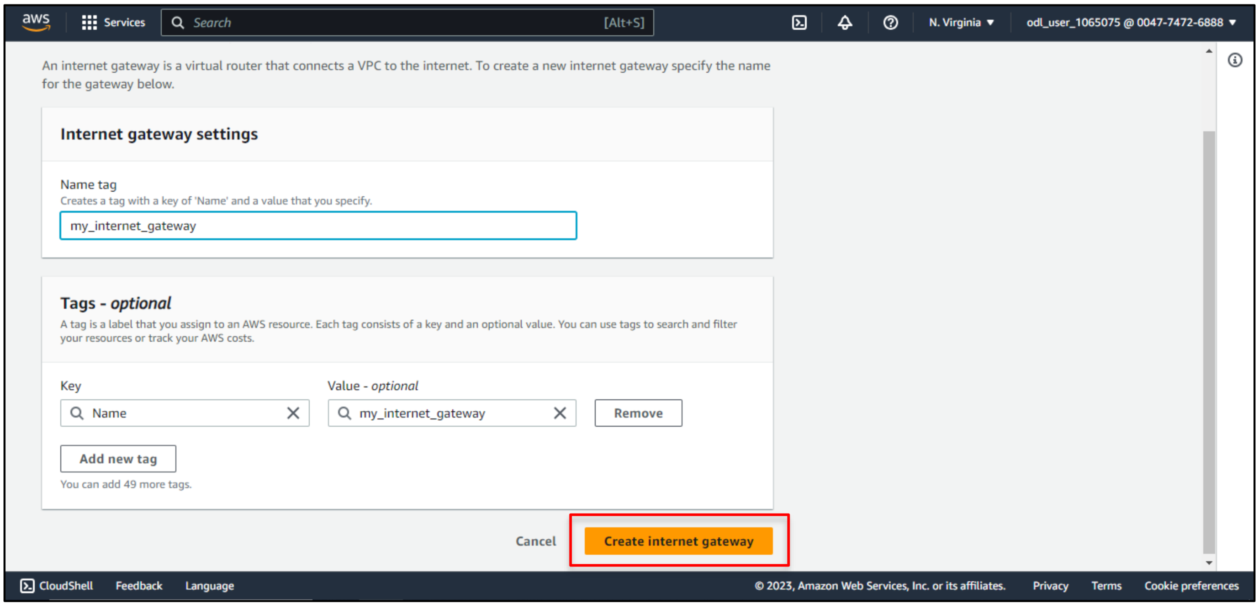


Internet gateways (3) Info

Name	Internet gateway ID	State	VPC ID	Owner
my_internet_gateway	igw-07b562f5d99bf0c90	Attached	vpc-0c4914c53027e2ef3 | Simplilearn...	00477472
–	igw-087008ffd7436e1df	Attached	vpc-0c2694582c0e2a87e	00477472
my_internet_gateway	igw-09cec7d7ead2f1476	Detached	–	00477472

Select an internet gateway above

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



An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

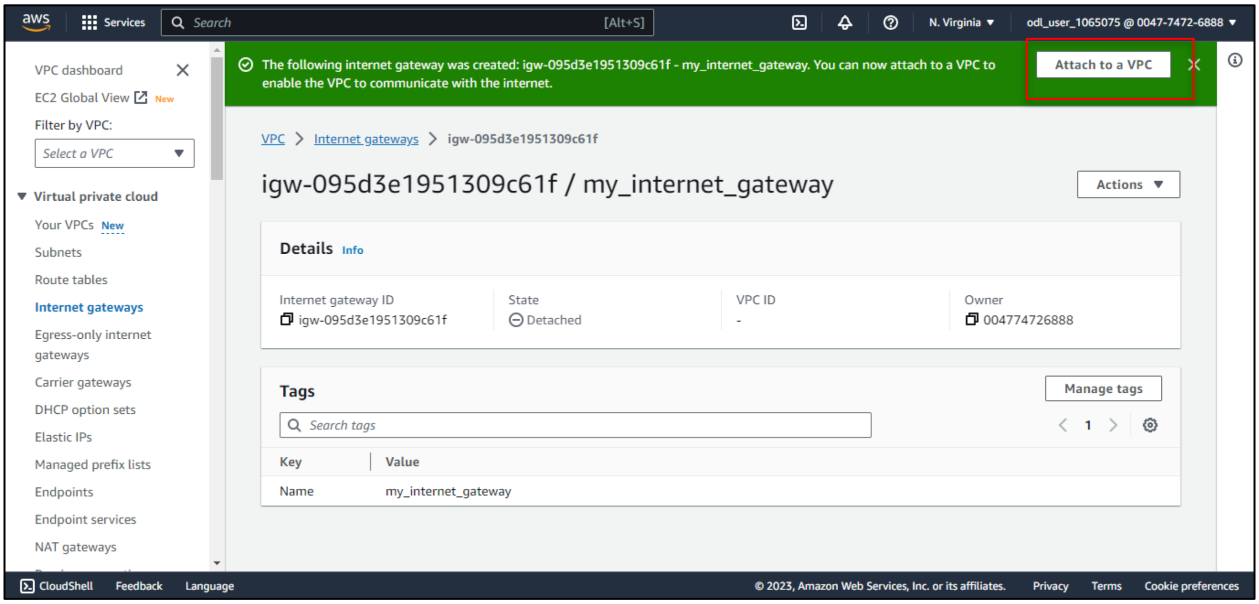
Name tag
Creates a tag with a key of 'Name' and a value that you specify.
my_internet_gateway

Tags - *optional*
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	my_internet_gateway

You can add 49 more tags.

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



The following internet gateway was created: igw-095d3e1951309c61f - my_internet_gateway. You can now attach to a VPC to enable the VPC to communicate with the internet.

VPC > Internet gateways > igw-095d3e1951309c61f

igw-095d3e1951309c61f / my_internet_gateway

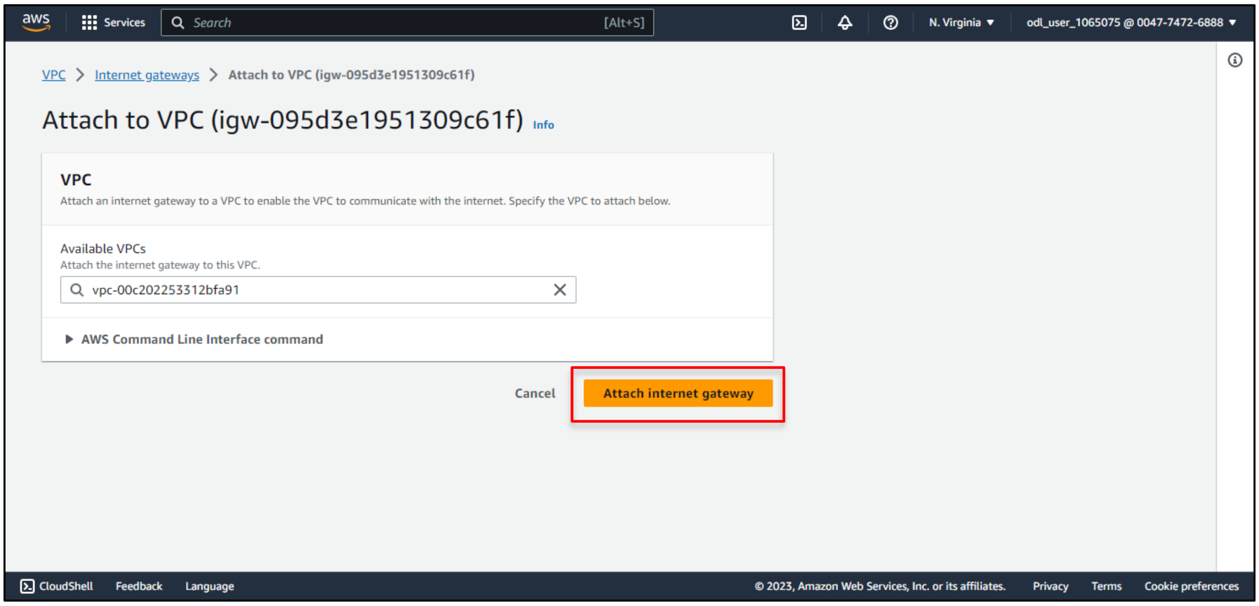
Details Info

Internet gateway ID	State	VPC ID	Owner
igw-095d3e1951309c61f	Detached	-	004774726888

Tags

Key	Value
Name	my_internet_gateway

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



aws
Services
Search [Alt+S]
N. Virginia
odl_user_1065075 @ 0047-7472-6888
VPC > Internet gateways > Attach to VPC (igw-095d3e1951309c61f)
Cancel
Attach internet gateway
CloudShell Feedback Language
© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Attach to VPC (igw-095d3e1951309c61f) Info

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

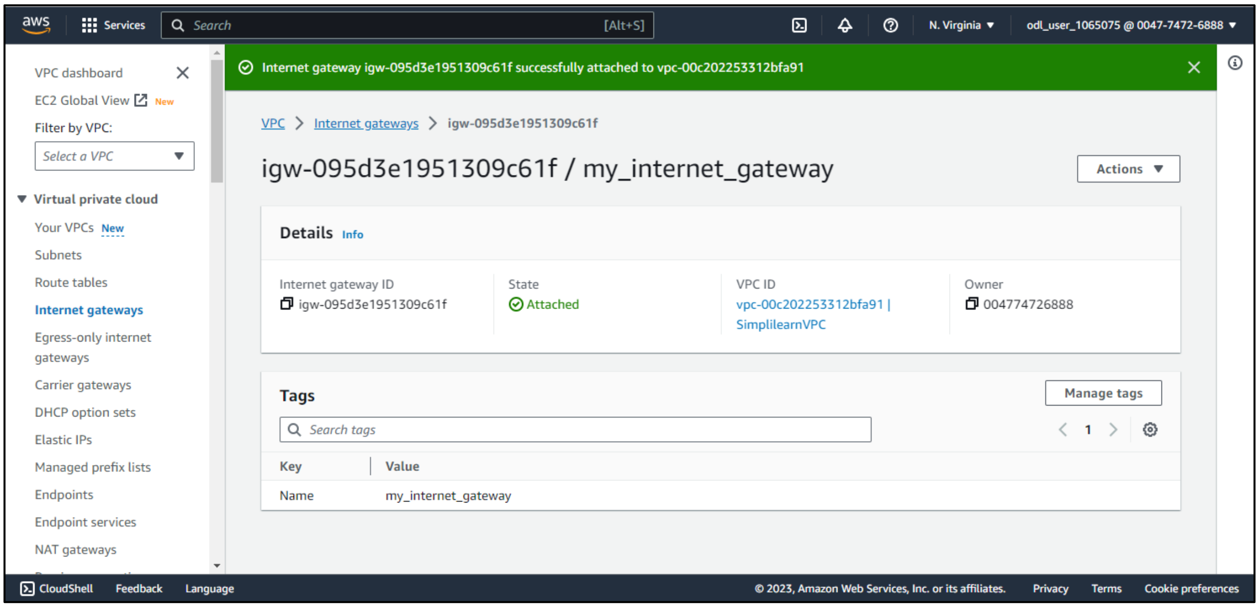
Available VPCs

Attach the internet gateway to this VPC.

vpc-00c202253312bfa91

▶ AWS Command Line Interface command

The following screen will appear:



Internet gateway igw-095d3e1951309c61f successfully attached to vpc-00c202253312bfa91

igw-095d3e1951309c61f / my_internet_gateway

Details Info

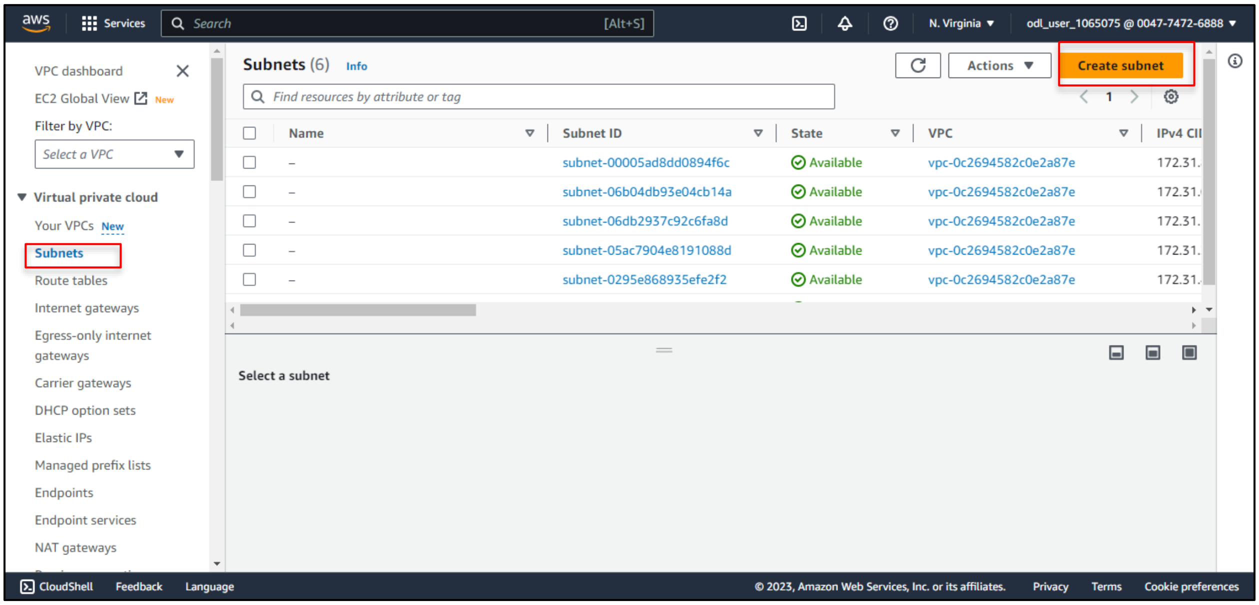
Internet gateway ID	State	VPC ID	Owner
igw-095d3e1951309c61f	Attached	vpc-00c202253312bfa91 | SimplilearnVPC	004774726888

Tags

Key	Value
Name	my_internet_gateway

Step 3: Create three subnets

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

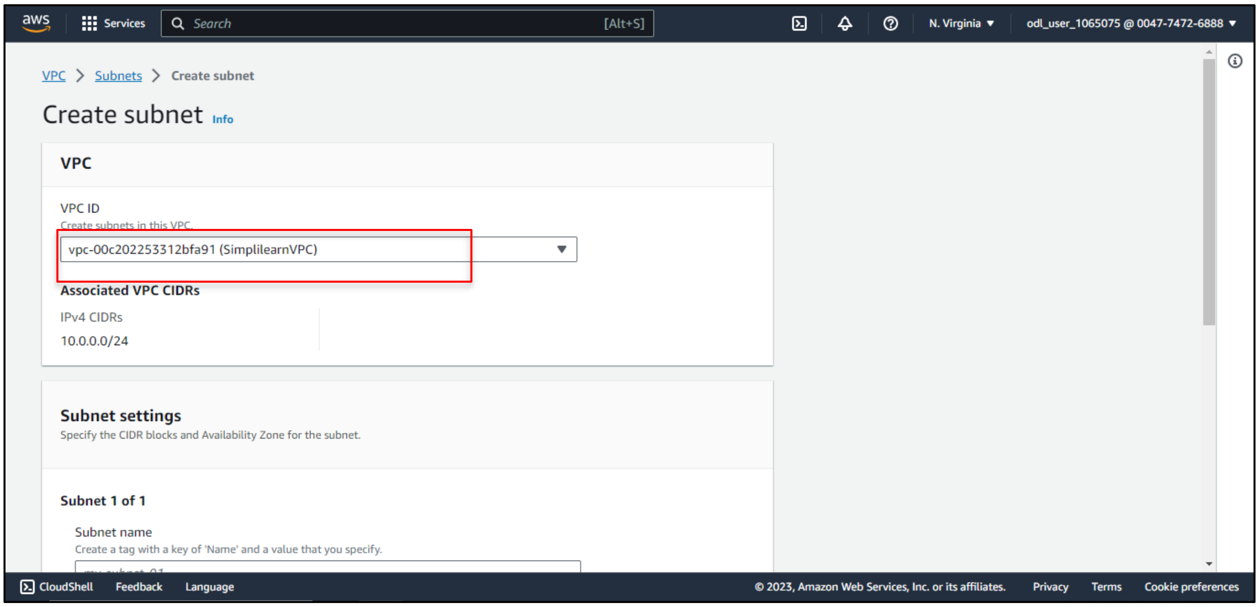


Subnets (6)

	Name	Subnet ID	State	VPC	IPv4 CII
□	–	subnet-00005ad8dd0894f6c	Available	vpc-0c2694582c0e2a87e	172.31.
□	–	subnet-06b04db93e04cb14a	Available	vpc-0c2694582c0e2a87e	172.31.
□	–	subnet-06db2937c92c6fa8d	Available	vpc-0c2694582c0e2a87e	172.31.
□	–	subnet-05ac7904e8191088d	Available	vpc-0c2694582c0e2a87e	172.31.
□	–	subnet-0295e868935efe2f2	Available	vpc-0c2694582c0e2a87e	172.31.

Select a subnet

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



Create subnet

VPC

VPC ID
Create subnets in this VPC.
vpc-00c202253312bfa91 (SimplilearnVPC)

Associated VPC CIDRs

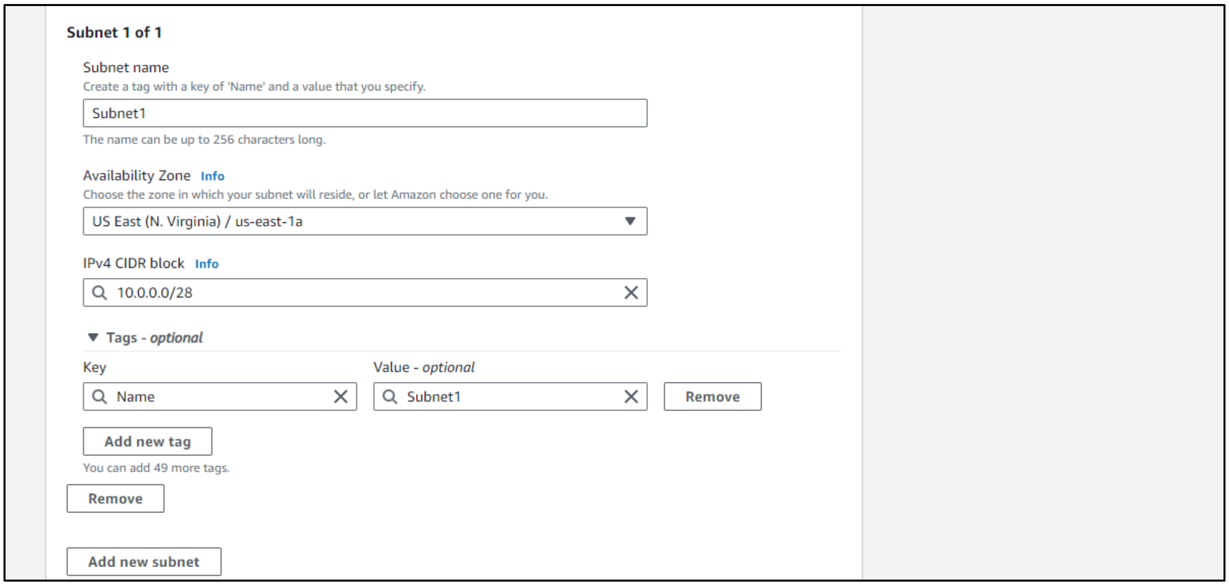
IPv4 CIDRs
10.0.0.0/24

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.

3.3 In the Subnet settings section, create three subnets by repeating earlier steps with specified settings: one in the **us-east-1a** zone with a **10.0.0.0/28** CIDR block, another in **us-east-1b** with a **10.0.0.16/28** CIDR block, and a third in **us-east-1c** with a **10.0.0.32/28** CIDR block.



Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
Subnet1
The name can be up to 256 characters long.

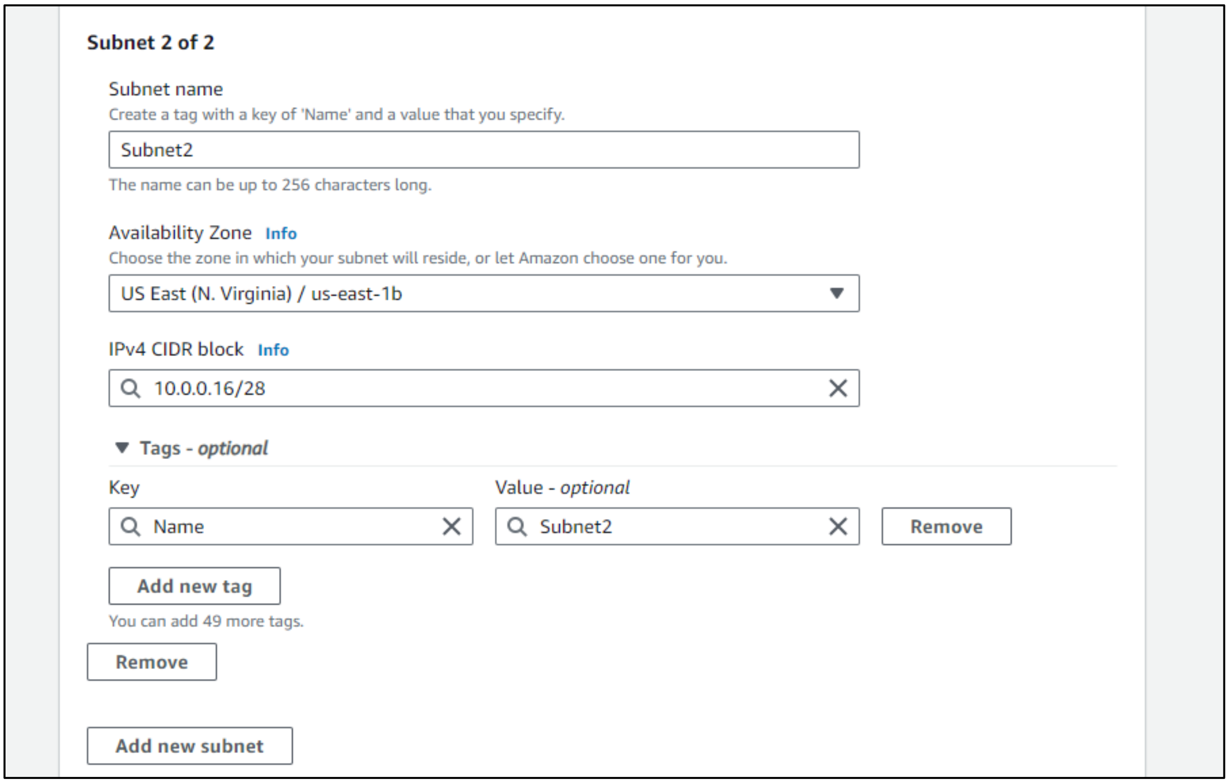
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1a

IPv4 CIDR block Info
10.0.0.0/28

Tags - *optional*

Key	Value - *optional*
Name	Subnet1

You can add 49 more tags.



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.

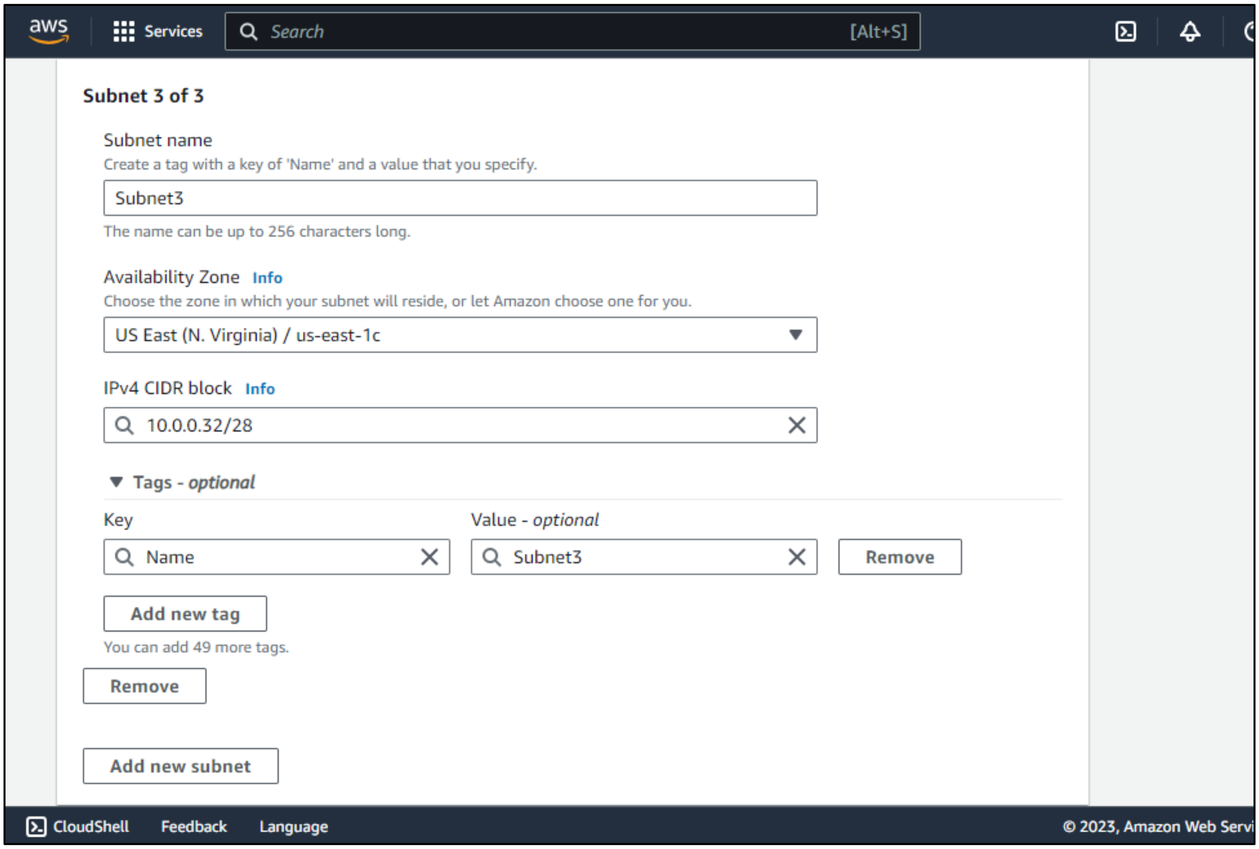
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1b

IPv4 CIDR block Info
10.0.0.16/28

Tags - *optional*

Key	Value - *optional*
Name	Subnet2

You can add 49 more tags.



Subnet 3 of 3

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
Subnet3
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-1c

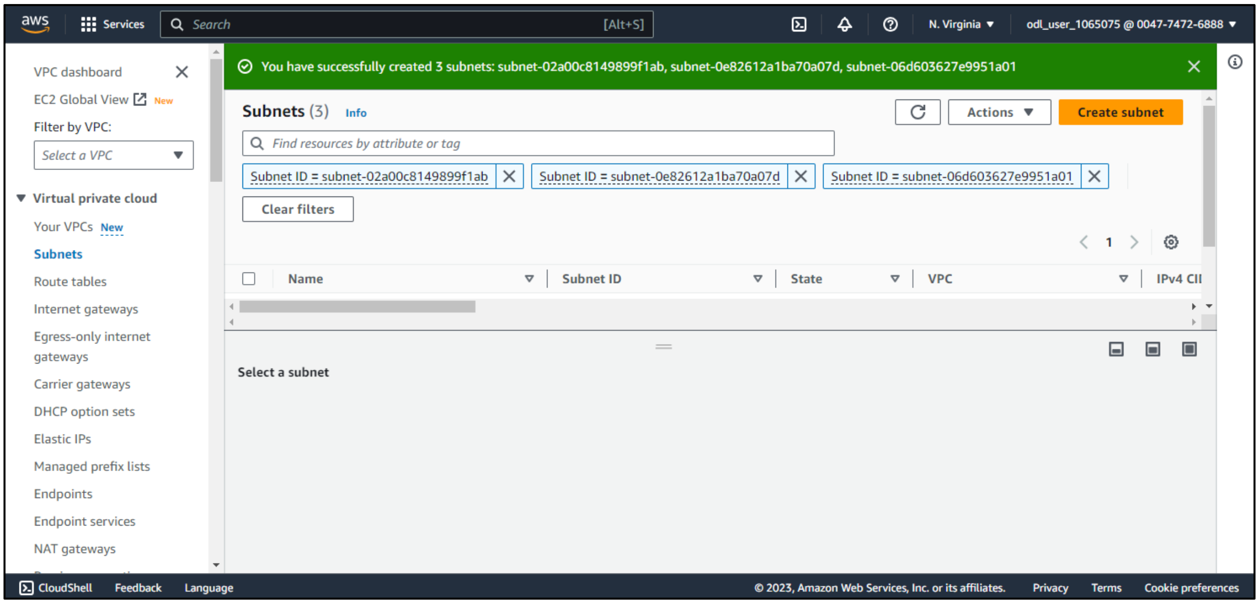
IPv4 CIDR block Info
10.0.0.32/28

Tags - *optional*

Key	Value - *optional*
Name	Subnet3

You can add 49 more tags.

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



You have successfully created 3 subnets: subnet-02a00c8149899f1ab, subnet-0e82612a1ba70a07d, subnet-06d603627e9951a01

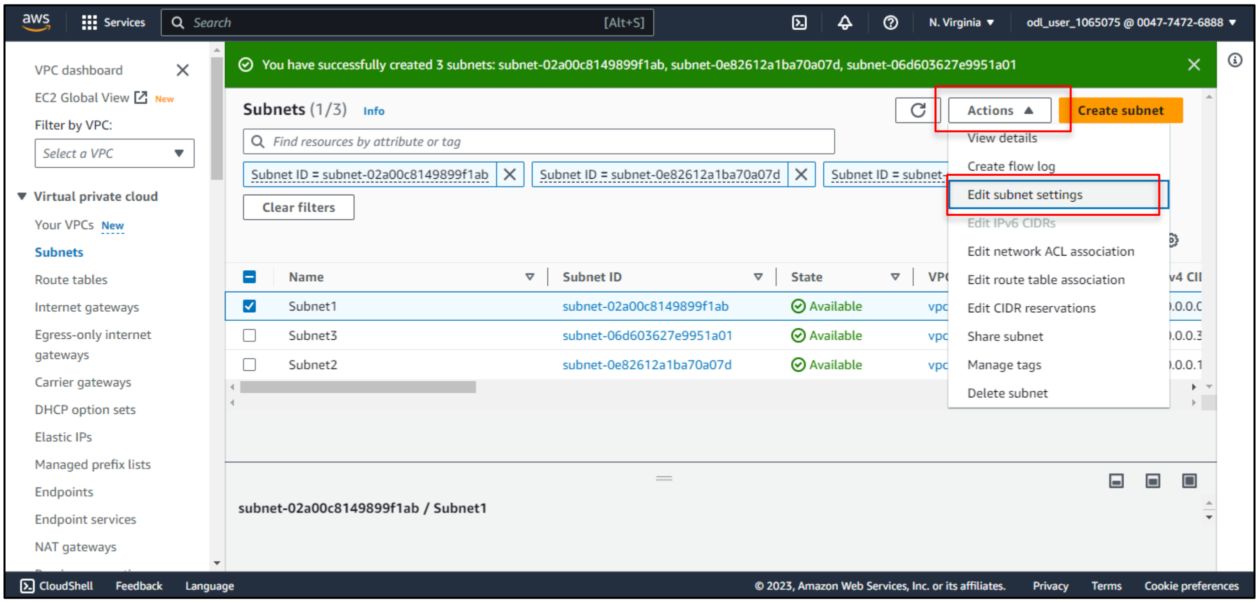
Subnets (3) Info

Subnet ID = subnet-02a00c8149899f1ab
Subnet ID = subnet-0e82612a1ba70a07d
Subnet ID = subnet-06d603627e9951a01

Name	Subnet ID	State	VPC	IPv4 CII

Select a subnet

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



You have successfully created 3 subnets: subnet-02a00c8149899f1ab, subnet-0e82612a1ba70a07d, subnet-06d603627e9951a01

Subnets (1/3) Info

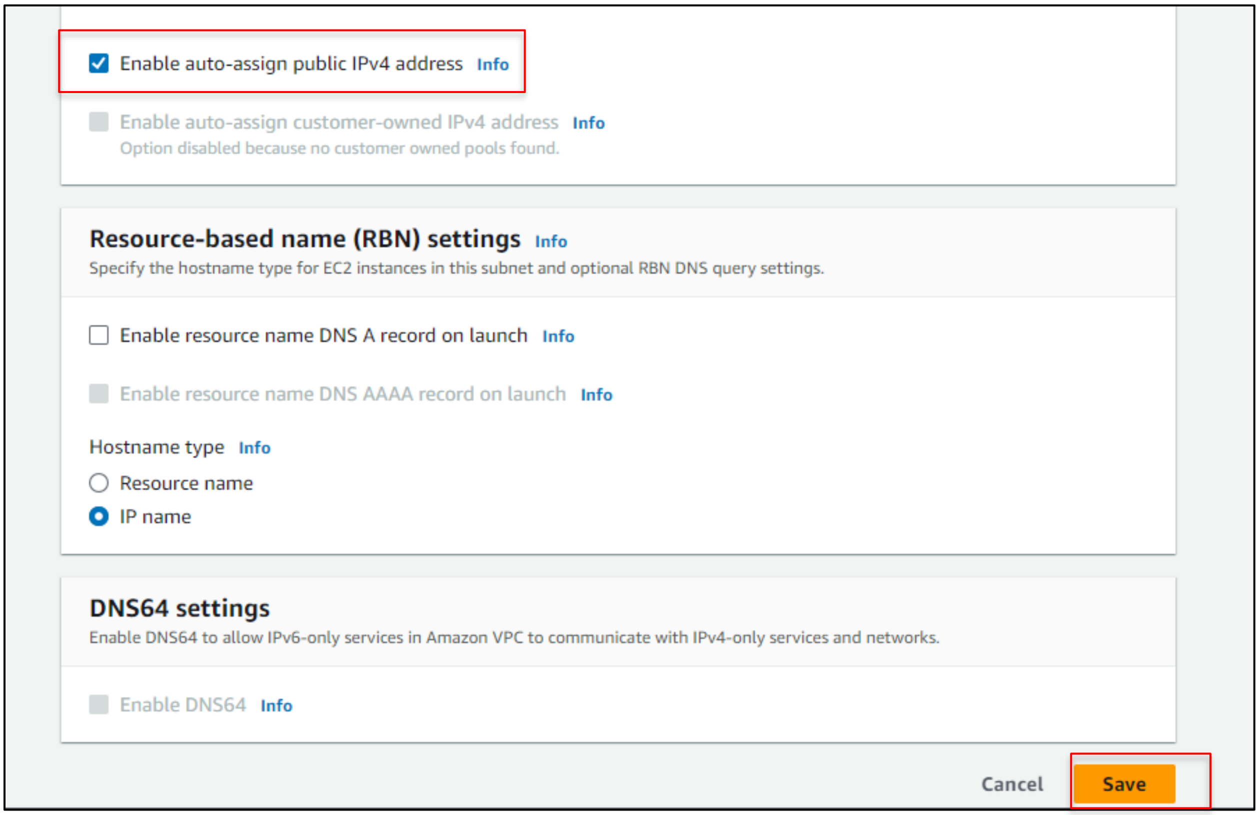
Find resources by attribute or tag

Subnet ID = subnet-02a00c8149899f1ab | Subnet ID = subnet-0e82612a1ba70a07d | Subnet ID = subnet-

	Name	Subnet ID	State	VPC
☑	Subnet1	subnet-02a00c8149899f1ab	Available	vpc
□	Subnet3	subnet-06d603627e9951a01	Available	vpc
□	Subnet2	subnet-0e82612a1ba70a07d	Available	vpc

subnet-02a00c8149899f1ab / Subnet1

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



☑ Enable auto-assign public IPv4 address Info

□ Enable auto-assign customer-owned IPv4 address Info
Option disabled because no customer owned pools found.

Resource-based name (RBN) settings Info

Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.

□ Enable resource name DNS A record on launch Info

□ Enable resource name DNS AAAA record on launch Info

Hostname type Info

○ Resource name

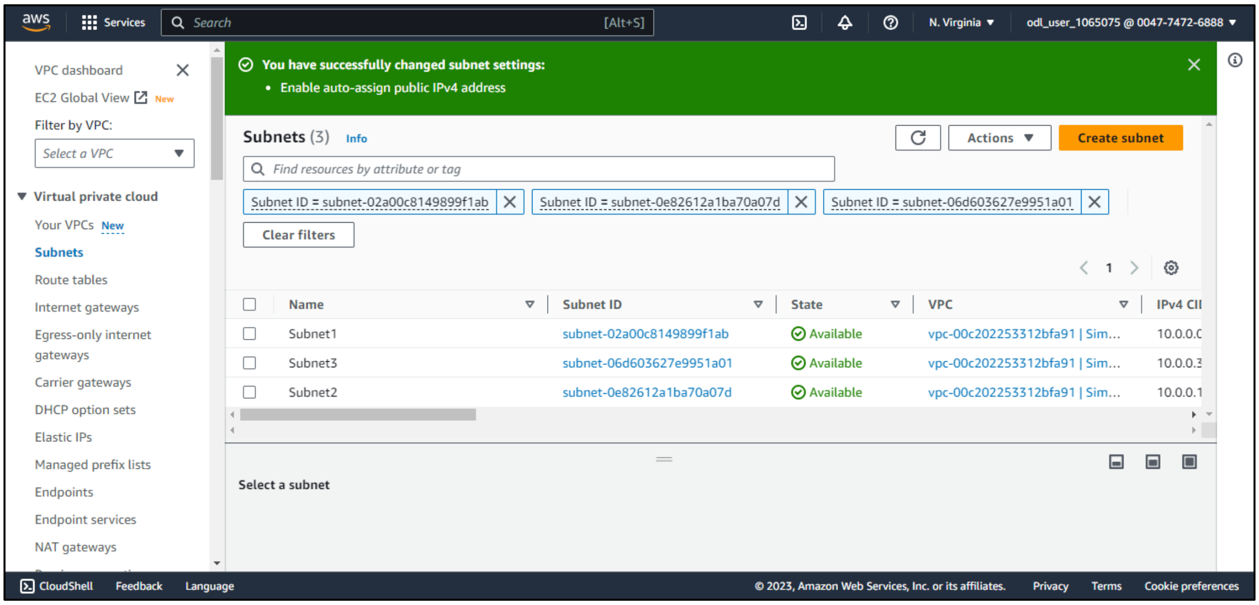
● IP name

DNS64 settings

Enable DNS64 to allow IPv6-only services in Amazon VPC to communicate with IPv4-only services and networks.

□ Enable DNS64 Info

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



You have successfully changed subnet settings:

- Enable auto-assign public IPv4 address

Subnets (3) Info

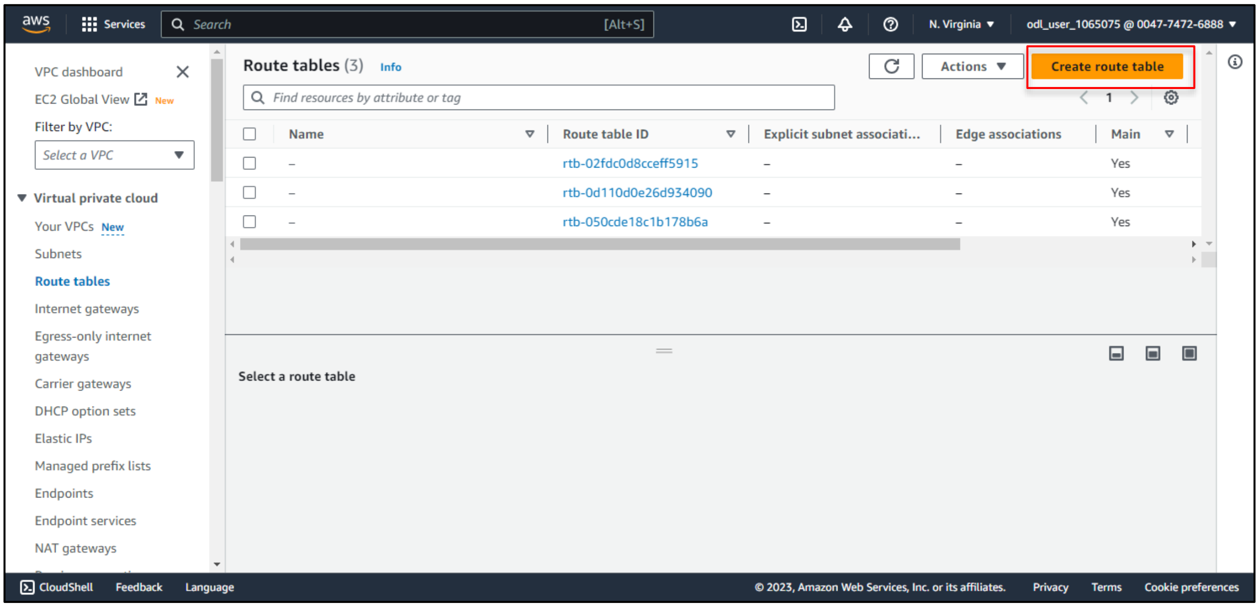
Subnet ID = subnet-02a00c8149899f1ab | Subnet ID = subnet-0e82612a1ba70a07d | Subnet ID = subnet-06d603627e9951a01

Name	Subnet ID	State	VPC	IPv4 CII
Subnet1	subnet-02a00c8149899f1ab	Available	vpc-00c202253312bfa91 | Sim...	10.0.0.0
Subnet3	subnet-06d603627e9951a01	Available	vpc-00c202253312bfa91 | Sim...	10.0.0.3
Subnet2	subnet-0e82612a1ba70a07d	Available	vpc-00c202253312bfa91 | Sim...	10.0.0.1

Select a subnet

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**

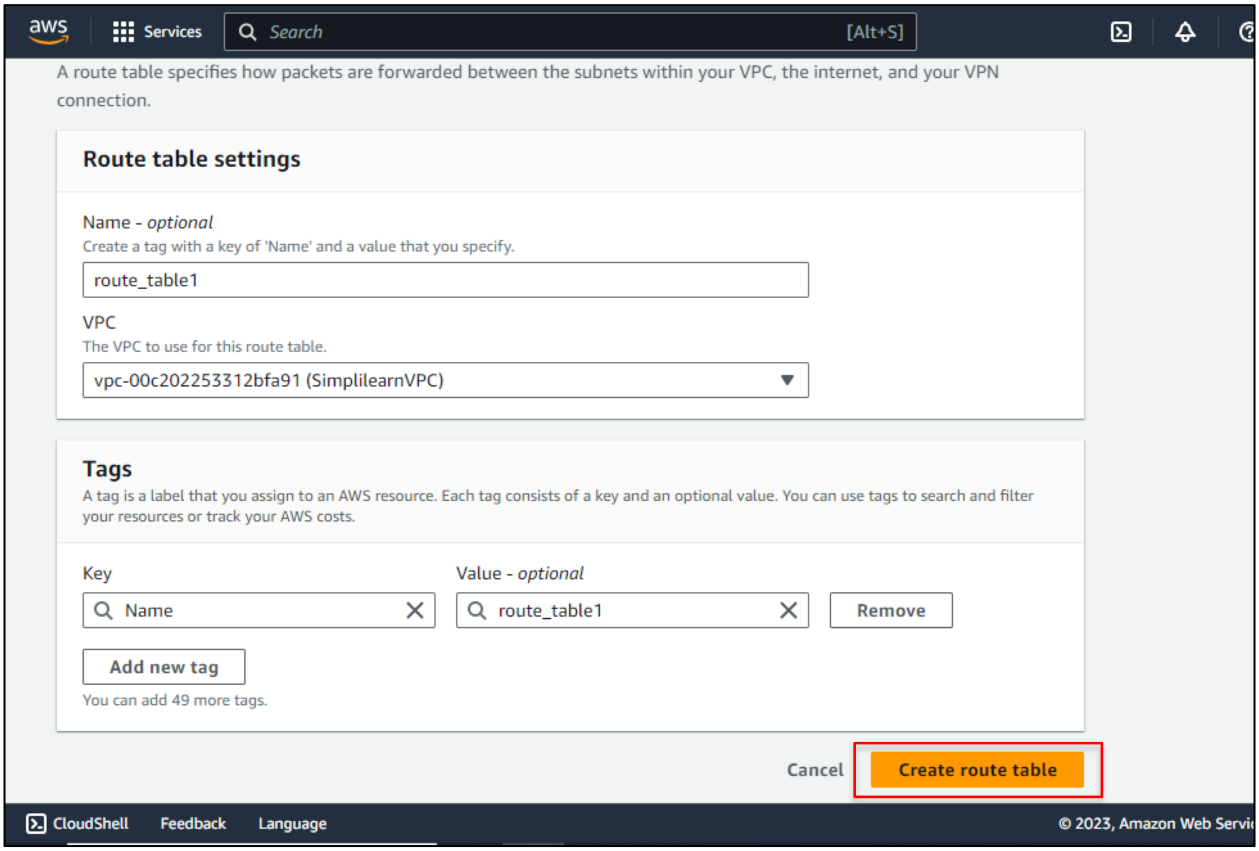


Route tables (3) Info

Name	Route table ID	Explicit subnet associati...	Edge associations	Main
–	rtb-02fdc0d8cceff5915	–	–	Yes
–	rtb-0d110d0e26d934090	–	–	Yes
–	rtb-050cde18c1b178b6a	–	–	Yes

Select a route table

4.2 Enter an arbitrary name for the route table and add the VPC created in Step 1 in the VPC section, Click on **Create route table**



A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*

Create a tag with a key of 'Name' and a value that you specify.

route_table1

VPC

The VPC to use for this route table.

vpc-00c202253312bfa91 (SimplilearnVPC)

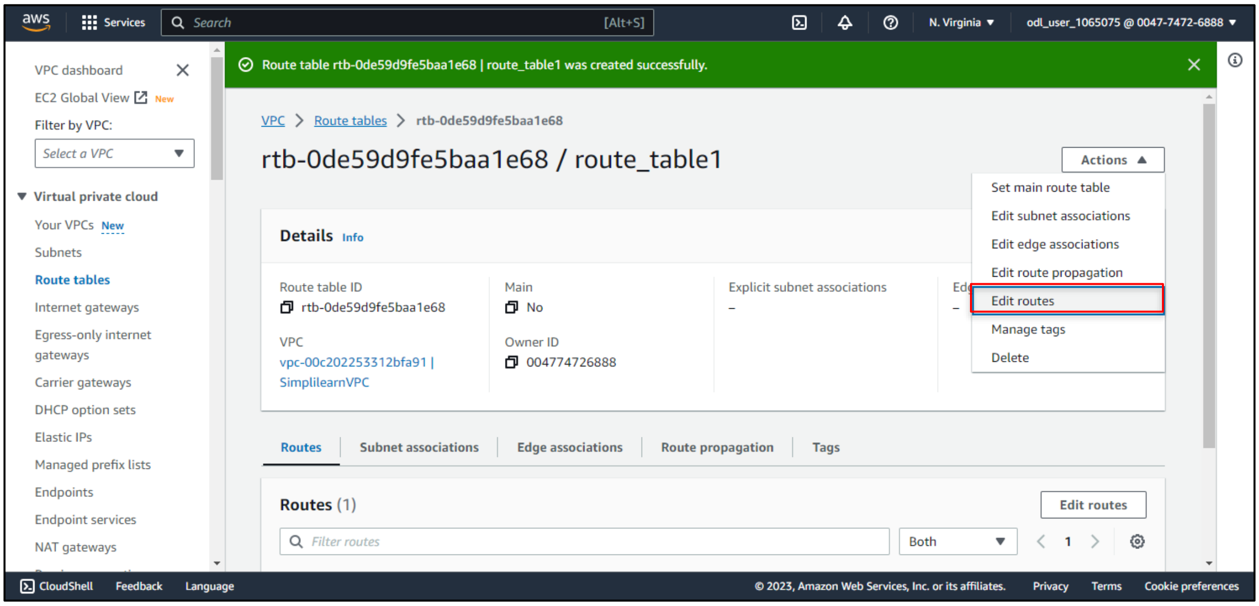
Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*
Name	route_table1

You can add 49 more tags.

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



Route table rtb-0de59d9fe5baa1e68 | route_table1 was created successfully.

VPC > Route tables > rtb-0de59d9fe5baa1e68

rtb-0de59d9fe5baa1e68 / route_table1

Details Info

Route table ID: rtb-0de59d9fe5baa1e68

Main: No

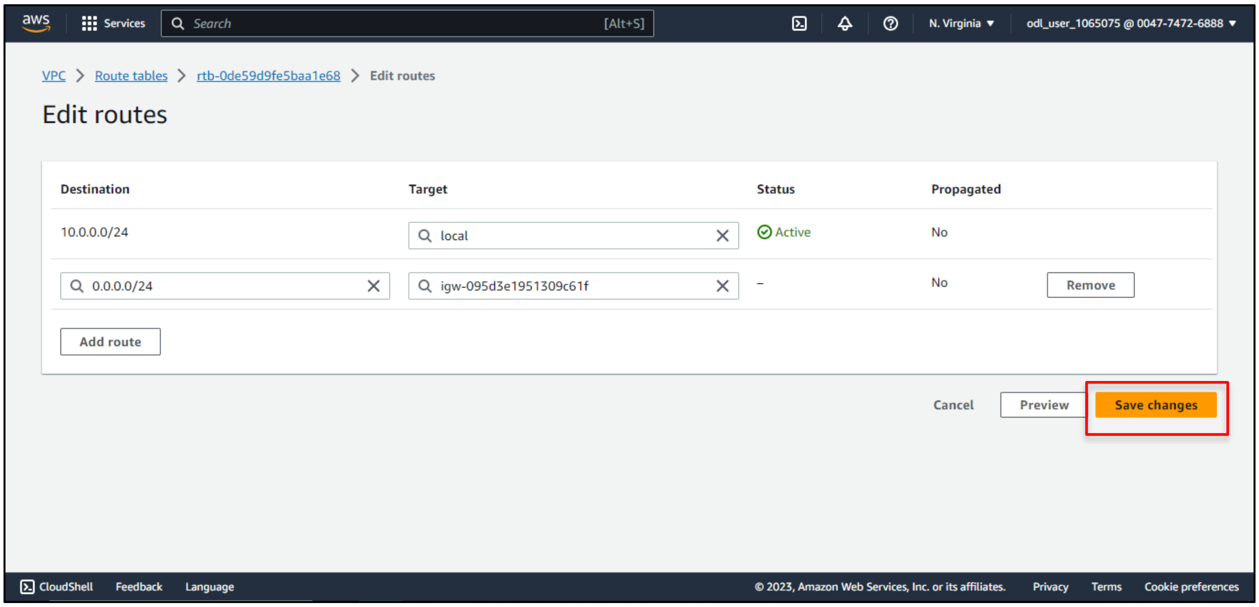
Explicit subnet associations: –

VPC: vpc-00c202253312bfa91 | SimplilearnVPC

Owner ID: 004774726888

Routes (1)

4.4 Select **Internet gateway (igw)** from the drop-down in the **Edit routes** window and Click on **Save changes**

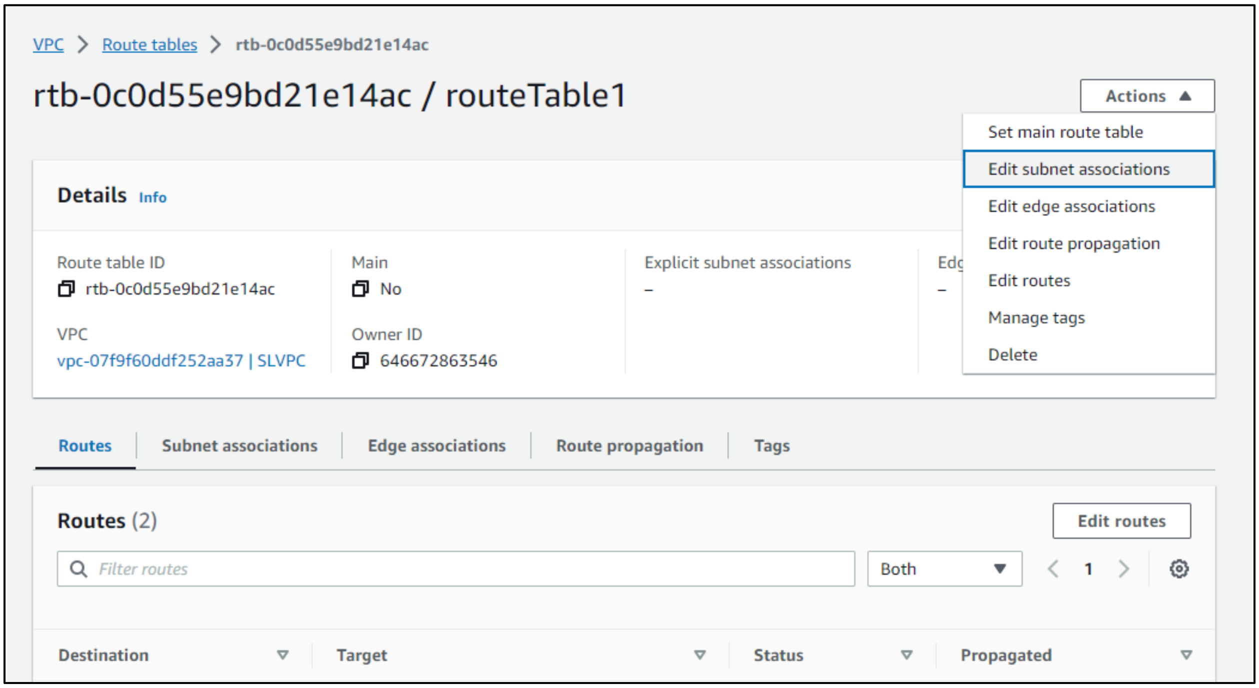


VPC > Route tables > rtb-0de59d9fe5baa1e68 > Edit routes

Edit routes

Destination	Target	Status	Propagated	
10.0.0.0/24	local	Active	No	
0.0.0.0/24	igw-095d3e1951309c61f	–	No	Remove

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



VPC > Route tables > rtb-0c0d55e9bd21e14ac

rtb-0c0d55e9bd21e14ac / routeTable1

Details Info

Route table ID: rtb-0c0d55e9bd21e14ac

Main: No

Explicit subnet associations: –

Edg –

VPC: vpc-07f9f60ddf252aa37 | SLVPC

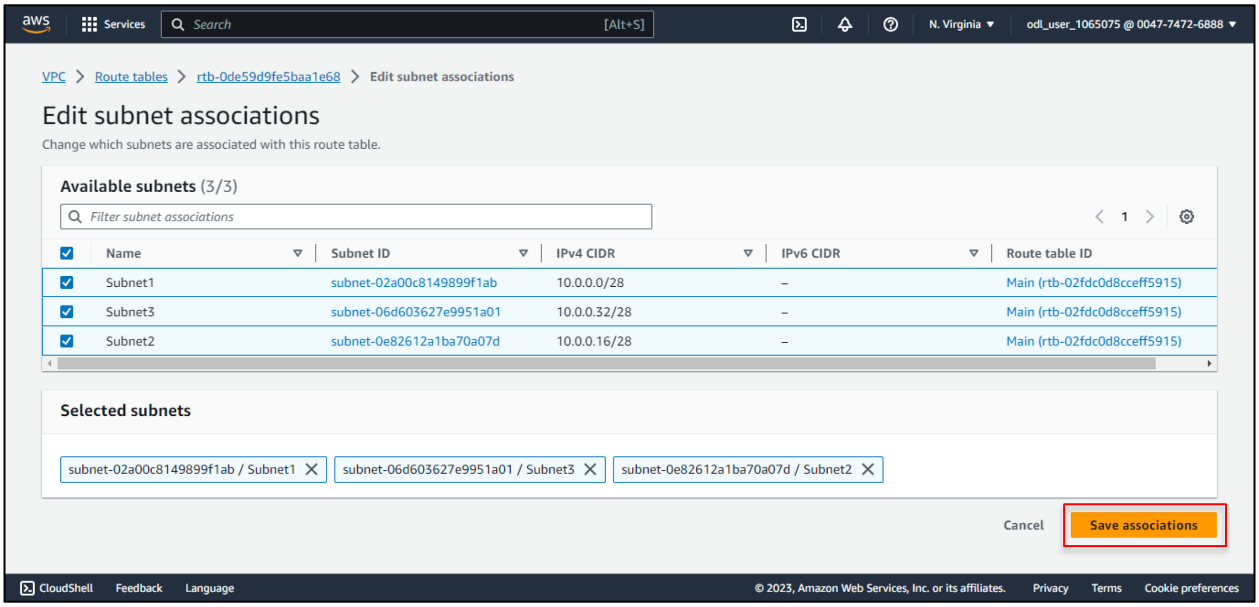
Owner ID: 646672863546

Routes | Subnet associations | Edge associations | Route propagation | Tags

Routes (2)

Destination	Target	Status	Propagated

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



aws
Services
Search [Alt+S]
N. Virginia ▼
odl_user_1065075 @ 0047-7472-6888 ▼

VPC > Route tables > rtb-0de59d9fe5baa1e68 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (3/3)

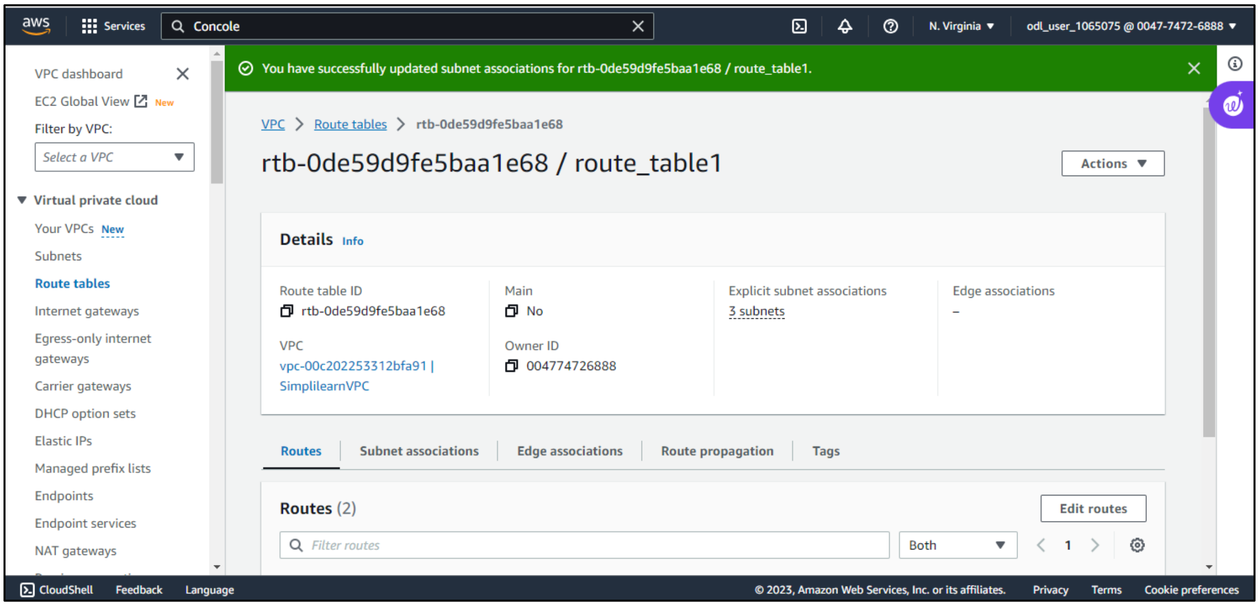
Filter subnet associations
< 1 >

☑	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
☑	Subnet1	subnet-02a00c8149899f1ab	10.0.0.0/28	–	Main (rtb-02fdc0d8cceff5915)
☑	Subnet3	subnet-06d603627e9951a01	10.0.0.32/28	–	Main (rtb-02fdc0d8cceff5915)
☑	Subnet2	subnet-0e82612a1ba70a07d	10.0.0.16/28	–	Main (rtb-02fdc0d8cceff5915)

Selected subnets

subnet-02a00c8149899f1ab / Subnet1 ✕ subnet-06d603627e9951a01 / Subnet3 ✕ subnet-0e82612a1ba70a07d / Subnet2 ✕

The following screen will appear:



aws Services Console N. Virginia odl_user_1065075 @ 0047-7472-6888

You have successfully updated subnet associations for rtb-0de59d9fe5baa1e68 / route_table1.

VPC > Route tables > rtb-0de59d9fe5baa1e68

rtb-0de59d9fe5baa1e68 / route_table1 Actions

Details Info

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0de59d9fe5baa1e68	No	3 subnets	–
VPC	Owner ID		
vpc-00c202253312bfa91 | SimplilearnVPC	004774726888		

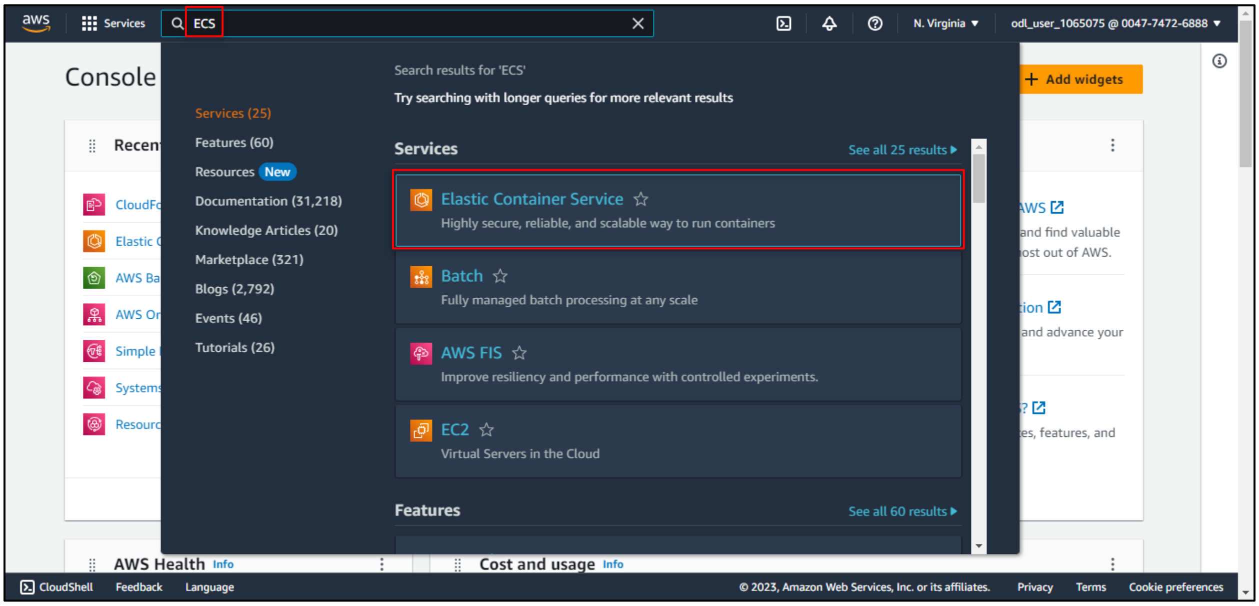
Routes | Subnet associations | Edge associations | Route propagation | Tags

Routes (2) Edit routes

Filter routes Both 1

Step 5: Create a cluster

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



aws Services ECS N. Virginia odl_user_1065075 @ 0047-7472-6888

Search results for 'ECS'

Try searching with longer queries for more relevant results

Services See all 25 results

Elastic Container Service
Highly secure, reliable, and scalable way to run containers

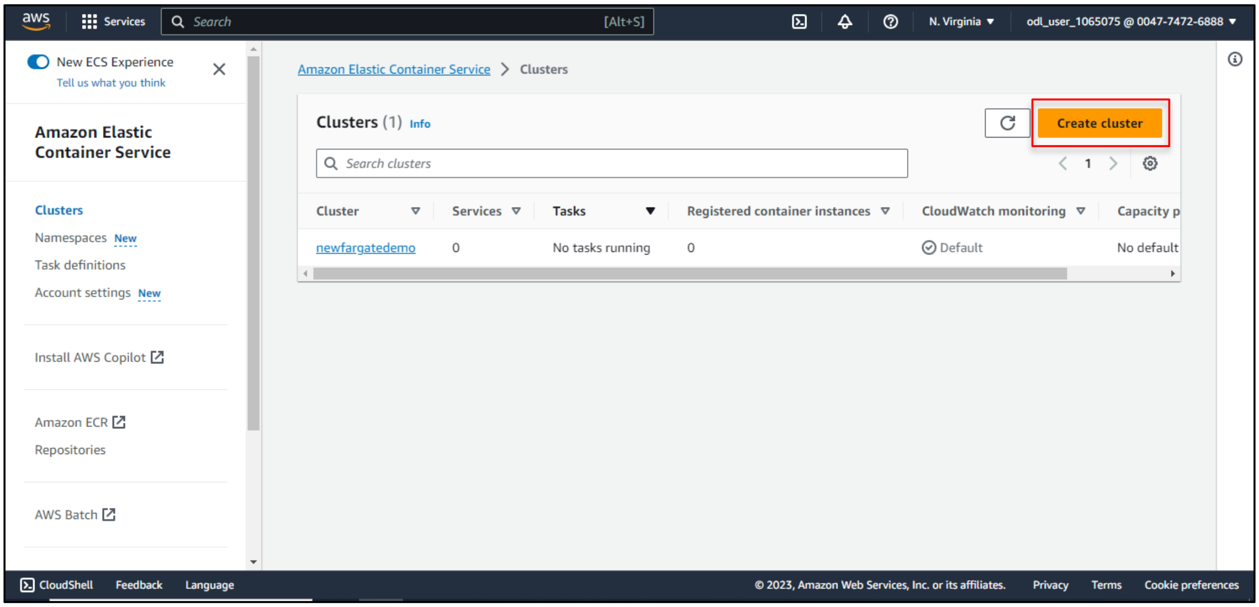
Batch
Fully managed batch processing at any scale

AWS FIS
Improve resiliency and performance with controlled experiments.

EC2
Virtual Servers in the Cloud

Features See all 60 results

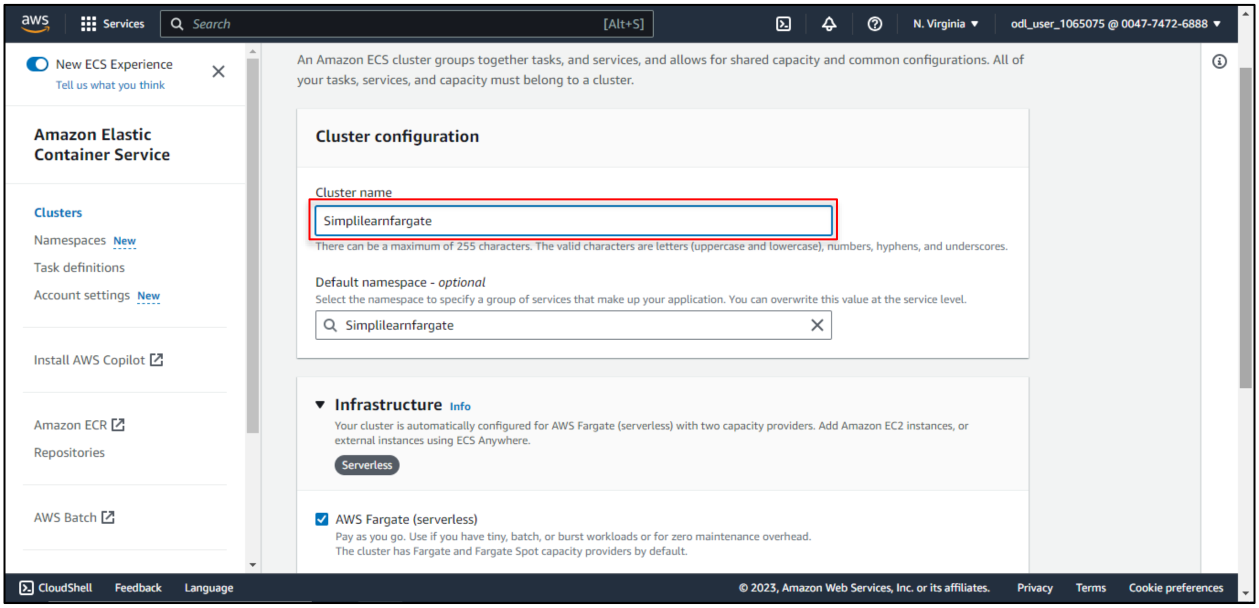
5.2 On the left panel of the ECS console, click on **Clusters** and **Create cluster**



Clusters (1) Info

Cluster	Services	Tasks	Registered container instances	CloudWatch monitoring	Capacity p
newfargatedemo	0	No tasks running	0	Default	No default

5.3 Enter an arbitrary name for the Cluster under the **Cluster name**



An Amazon ECS cluster groups together tasks, and services, and allows for shared capacity and common configurations. All of your tasks, services, and capacity must belong to a cluster.

Cluster configuration

Cluster name

Simplilearnfargate

There can be a maximum of 255 characters. The valid characters are letters (uppercase and lowercase), numbers, hyphens, and underscores.

Default namespace - *optional*

Select the namespace to specify a group of services that make up your application. You can overwrite this value at the service level.

Simplilearnfargate

Infrastructure Info

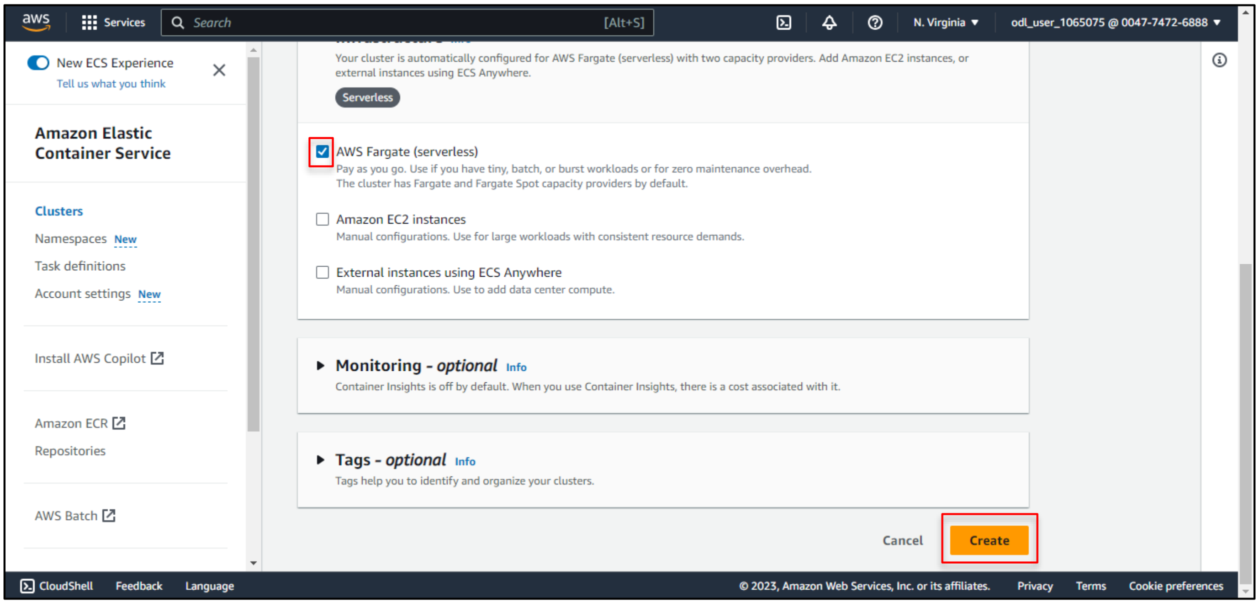
Your cluster is automatically configured for AWS Fargate (serverless) with two capacity providers. Add Amazon EC2 instances, or external instances using ECS Anywhere.

Serverless

☑ AWS Fargate (serverless)

Pay as you go. Use if you have tiny, batch, or burst workloads or for zero maintenance overhead.
The cluster has Fargate and Fargate Spot capacity providers by default.

5.4 In the Infrastructure, specify **AWS Fargate (serverless),** leave other settings at their default values and click on **Create**



Your cluster is automatically configured for AWS Fargate (serverless) with two capacity providers. Add Amazon EC2 instances, or external instances using ECS Anywhere.

Serverless

☑ AWS Fargate (serverless)
Pay as you go. Use if you have tiny, batch, or burst workloads or for zero maintenance overhead.
The cluster has Fargate and Fargate Spot capacity providers by default.

□ Amazon EC2 instances
Manual configurations. Use for large workloads with consistent resource demands.

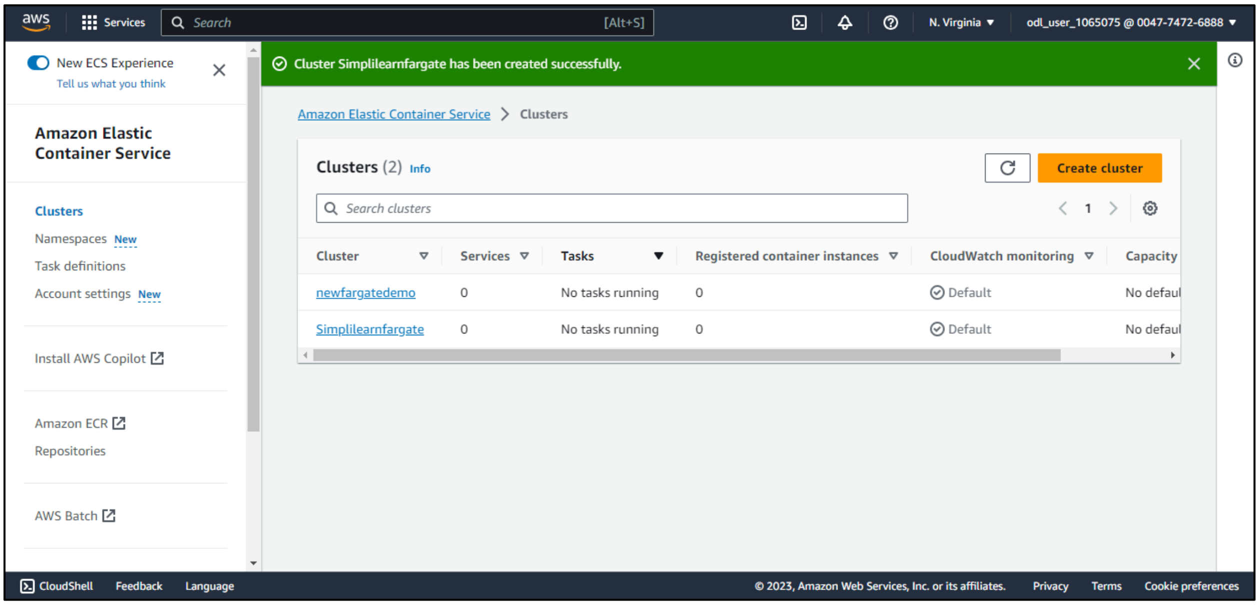
□ External instances using ECS Anywhere
Manual configurations. Use to add data center compute.

▸ Monitoring - *optional* Info
Container Insights is off by default. When you use Container Insights, there is a cost associated with it.

▸ Tags - *optional* Info
Tags help you to identify and organize your clusters.

Cancel | Create

5.5 Verify the cluster creation as shown below:



Cluster Simplilearnfargate has been created successfully.

Amazon Elastic Container Service > Clusters

Clusters (2) Info

Create cluster

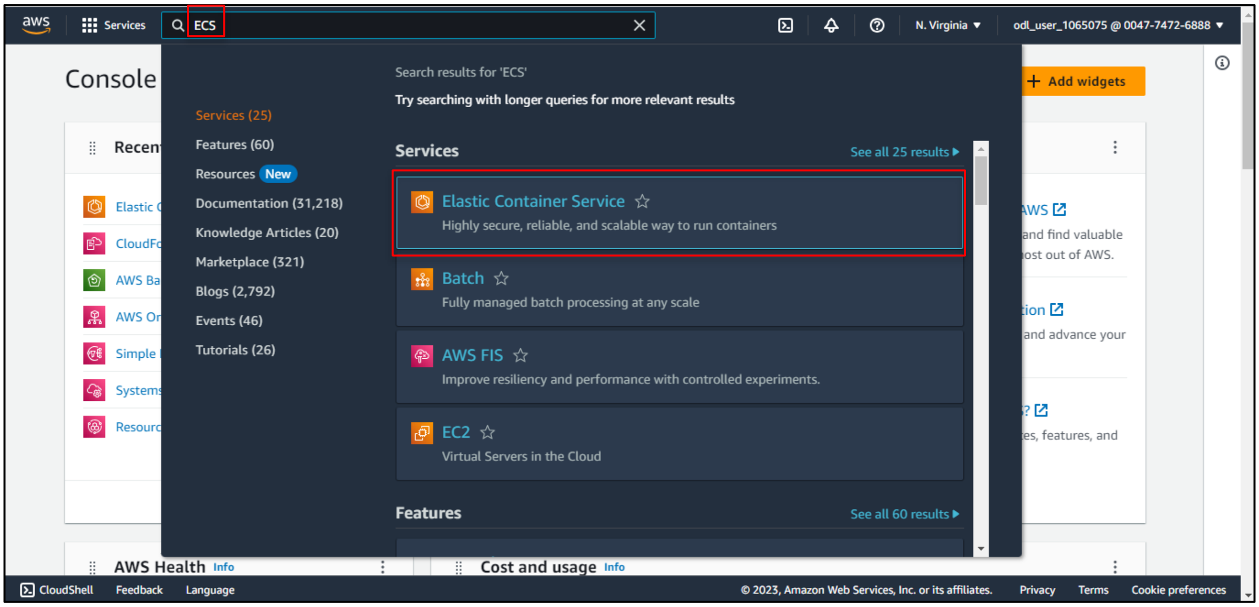
Search clusters

Cluster	Services	Tasks	Registered container instances	CloudWatch monitoring	Capacity
newfargatedemo	0	No tasks running	0	Default	No defaul
Simplilearnfargate	0	No tasks running	0	Default	No defaul

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

Step 6: Create Task definition

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



Search results for 'ECS'

Try searching with longer queries for more relevant results

Elastic Container Service
Highly secure, reliable, and scalable way to run containers

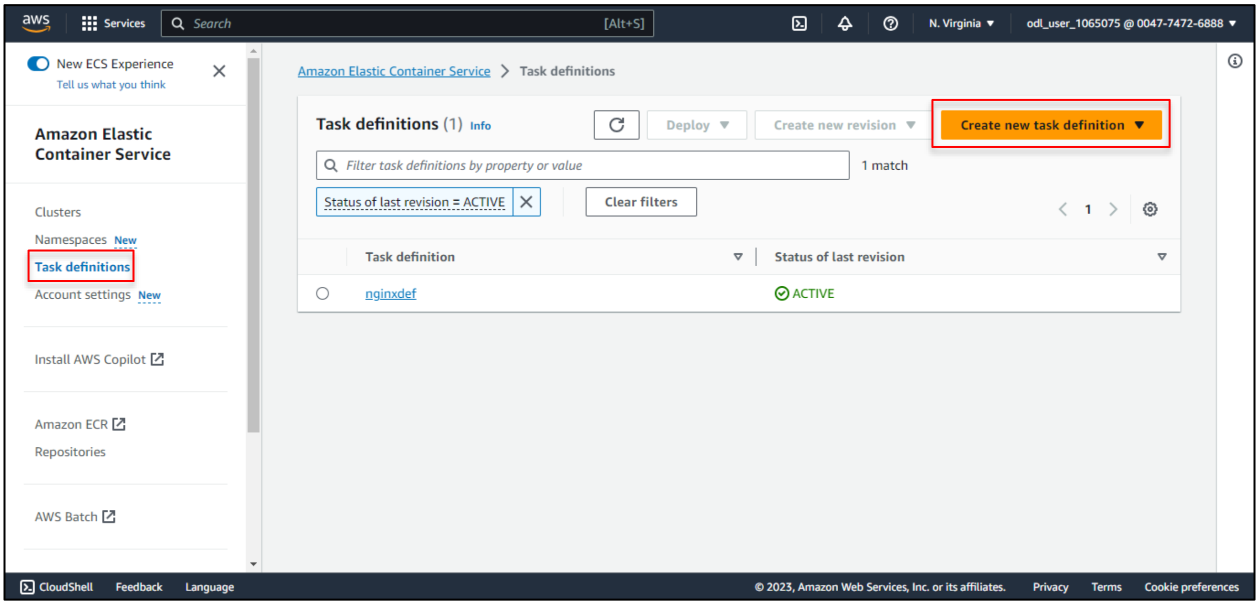
Batch
Fully managed batch processing at any scale

AWS FIS
Improve resiliency and performance with controlled experiments.

EC2
Virtual Servers in the Cloud

© 2023, Amazon Web Services, Inc. or its affiliates.

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

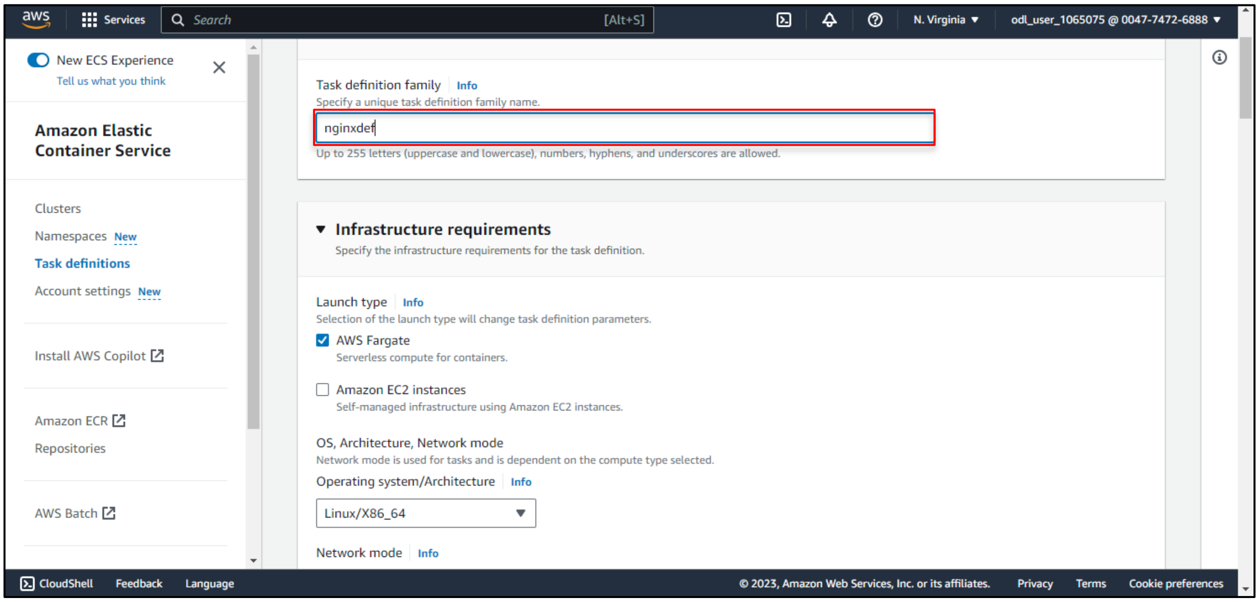


Task definitions (1)

Task definition	Status of last revision
nginxdef	ACTIVE

© 2023, Amazon Web Services, Inc. or its affiliates.

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



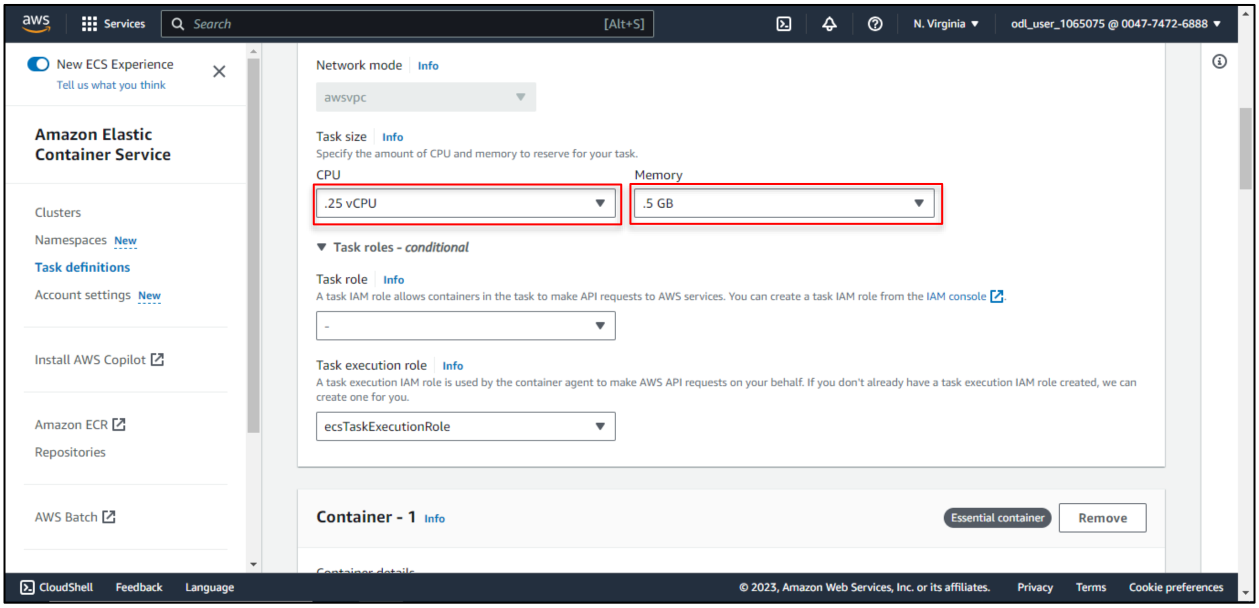
Task definition family Info
Specify a unique task definition family name.
nginxdef
Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

Infrastructure requirements
Specify the infrastructure requirements for the task definition.

Launch type Info
Selection of the launch type will change task definition parameters.
☑ AWS Fargate
Serverless compute for containers.
□ Amazon EC2 instances
Self-managed infrastructure using Amazon EC2 instances.

OS, Architecture, Network mode
Network mode is used for tasks and is dependent on the compute type selected.
Operating system/Architecture Info
Linux/X86_64

Network mode Info



Network mode Info
awsvpc

Task size Info
Specify the amount of CPU and memory to reserve for your task.
CPU
.25 vCPU
Memory
.5 GB

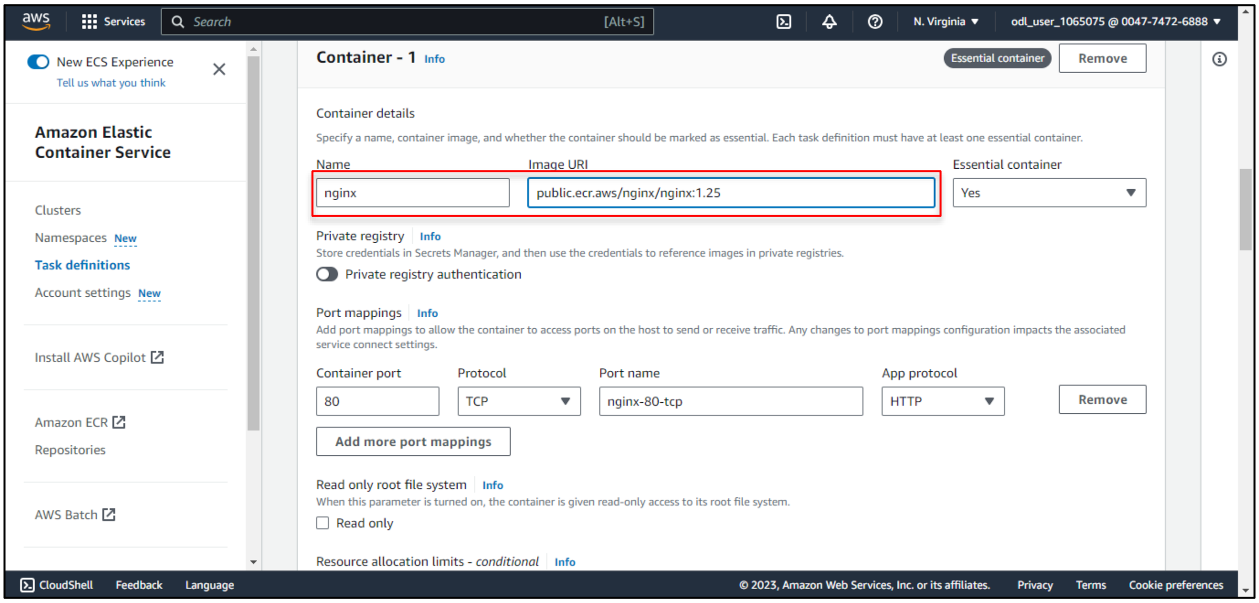
Task roles - *conditional*

Task role Info
A task IAM role allows containers in the task to make API requests to AWS services. You can create a task IAM role from the IAM console.
-

Task execution role Info
A task execution IAM role is used by the container agent to make AWS API requests on your behalf. If you don't already have a task execution IAM role created, we can create one for you.
ecsTaskExecutionRole

Container - 1 Info

6.4 For container - 1 details, enter **nginx** as name and **public.ecr.aws/nginx/nginx:1.25** as Image URI



Container - 1 Info

Container details

Specify a name, container image, and whether the container should be marked as essential. Each task definition must have at least one essential container.

Name	Image URI	Essential container
nginx	public.ecr.aws/nginx/nginx:1.25	Yes

Private registry Info

Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.

Private registry authentication

Port mappings Info

Add port mappings to allow the container to access ports on the host to send or receive traffic. Any changes to port mappings configuration impacts the associated service connect settings.

Container port	Protocol	Port name	App protocol
80	TCP	nginx-80-tcp	HTTP

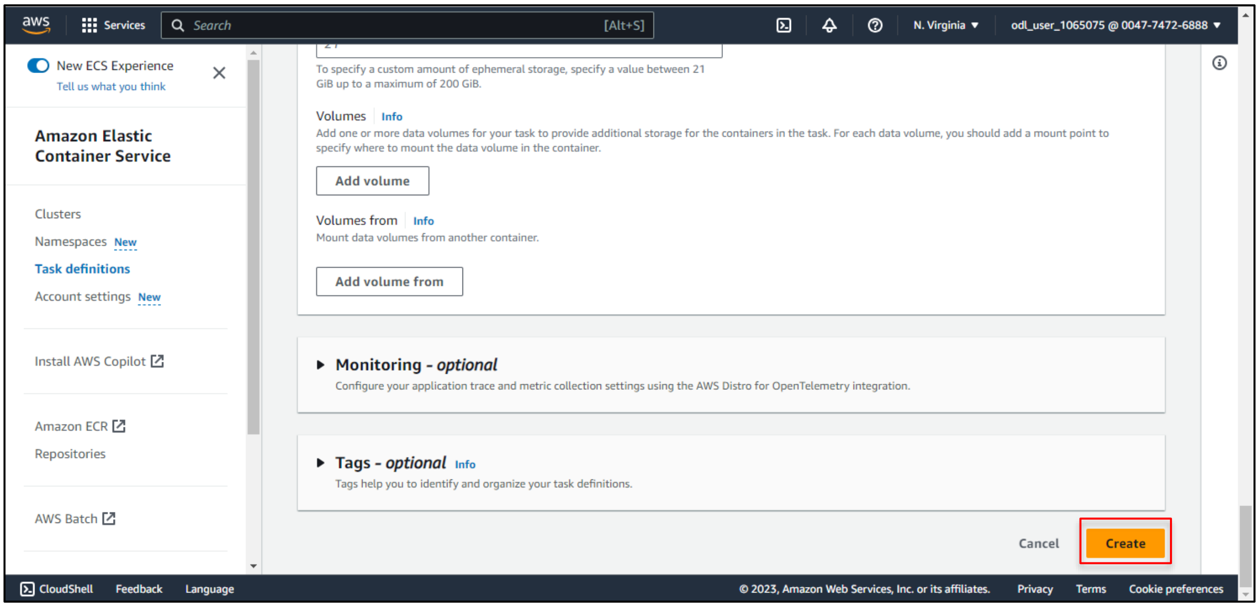
Read only root file system Info

When this parameter is turned on, the container is given read-only access to its root file system.

□ Read only

Resource allocation limits - *conditional* Info

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



To specify a custom amount of ephemeral storage, specify a value between 21 GiB up to a maximum of 200 GiB.

Volumes Info

Add one or more data volumes for your task to provide additional storage for the containers in the task. For each data volume, you should add a mount point to specify where to mount the data volume in the container.

Volumes from Info

Mount data volumes from another container.

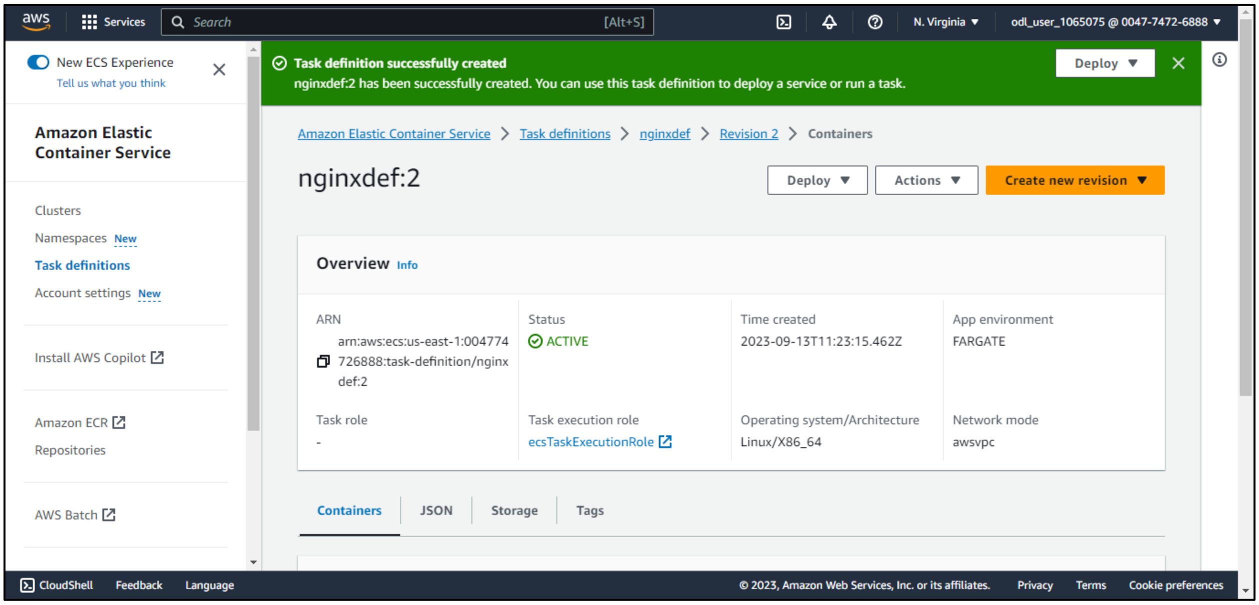
Monitoring - *optional*

Configure your application trace and metric collection settings using the AWS Distro for OpenTelemetry integration.

Tags - *optional* Info

Tags help you to identify and organize your task definitions.

Task definition has been successfully created as shown below:



Task definition successfully created

nginxdef:2 has been successfully created. You can use this task definition to deploy a service or run a task.

Amazon Elastic Container Service > Task definitions > nginxdef > Revision 2 > Containers

nginxdef:2

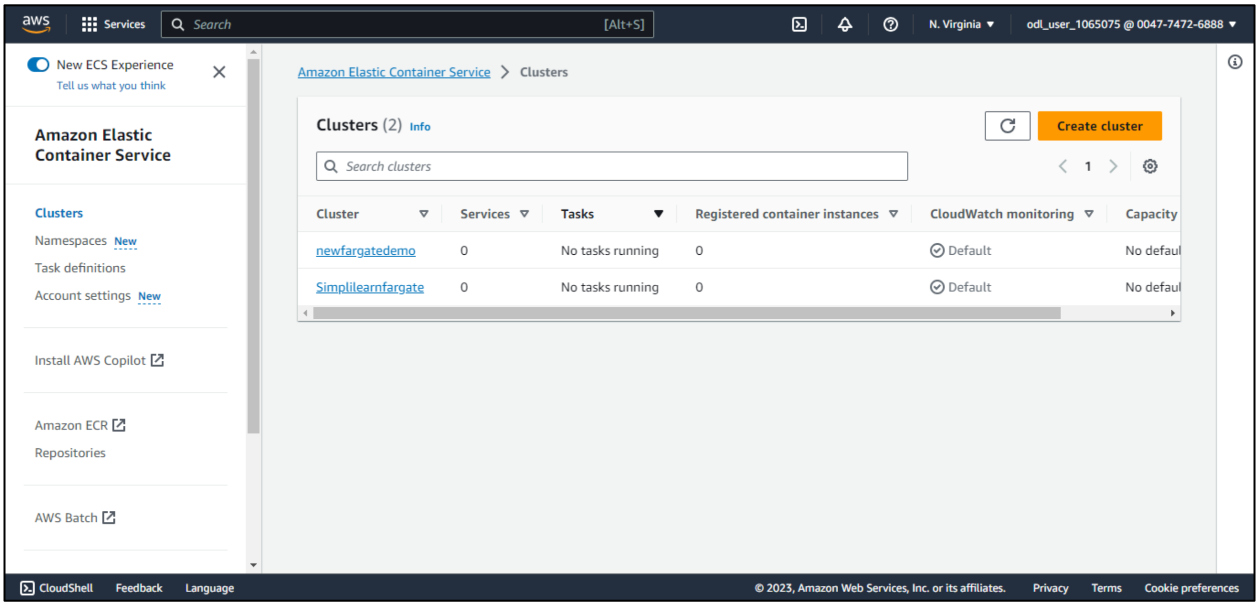
Overview Info

ARN	Status	Time created	App environment
arn:aws:ecs:us-east-1:004774726888:task-definition/nginxdef:2	ACTIVE	2023-09-13T11:23:15.462Z	FARGATE

Task role	Task execution role	Operating system/Architecture	Network mode
-	ecsTaskExecutionRole	Linux/X86_64	awsvpc

Step 7: Run the service on the cluster

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

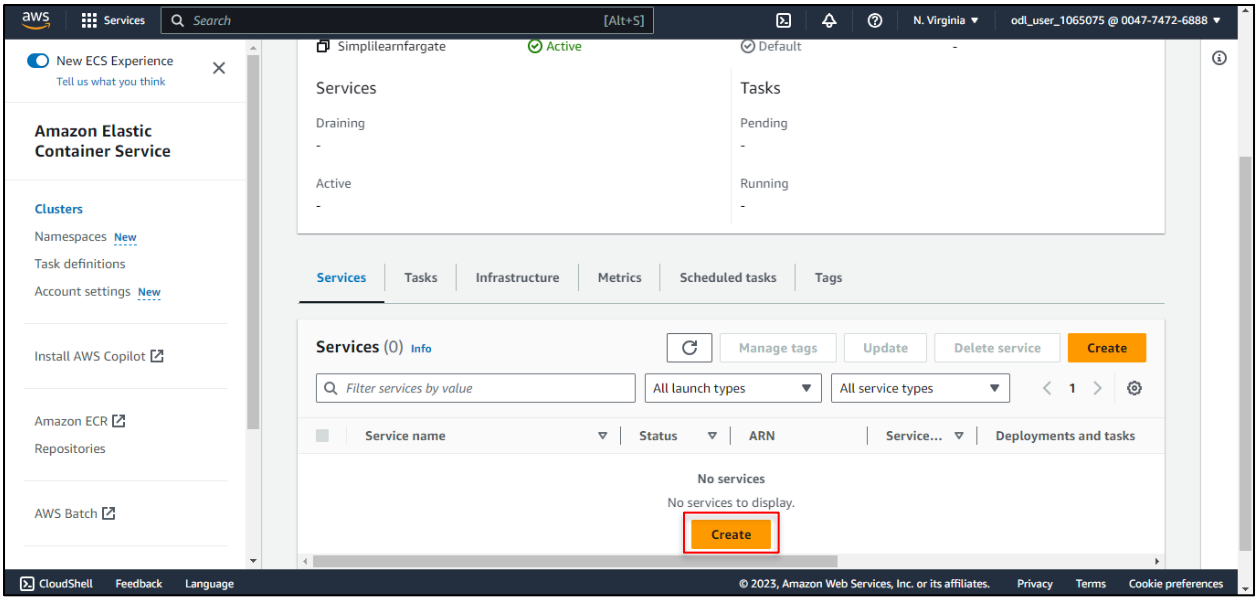


Amazon Elastic Container Service > Clusters

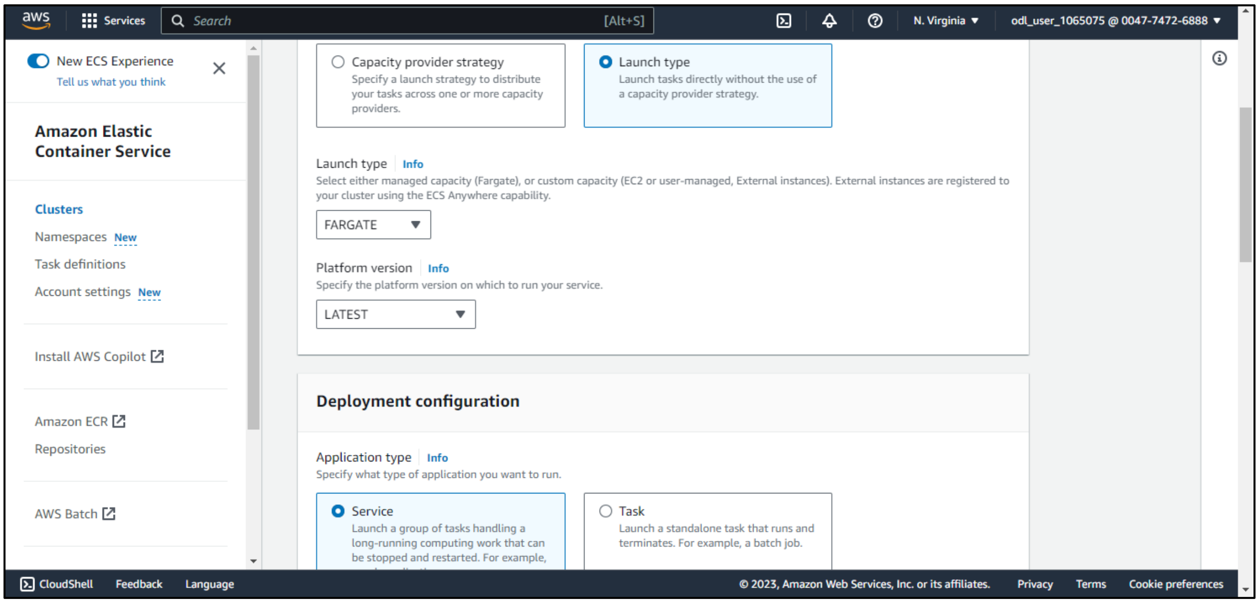
Clusters (2) Info

Cluster	Services	Tasks	Registered container instances	CloudWatch monitoring	Capacity
newfargatedemo	0	No tasks running	0	Default	No defaul
Simplilearnfargate	0	No tasks running	0	Default	No defaul

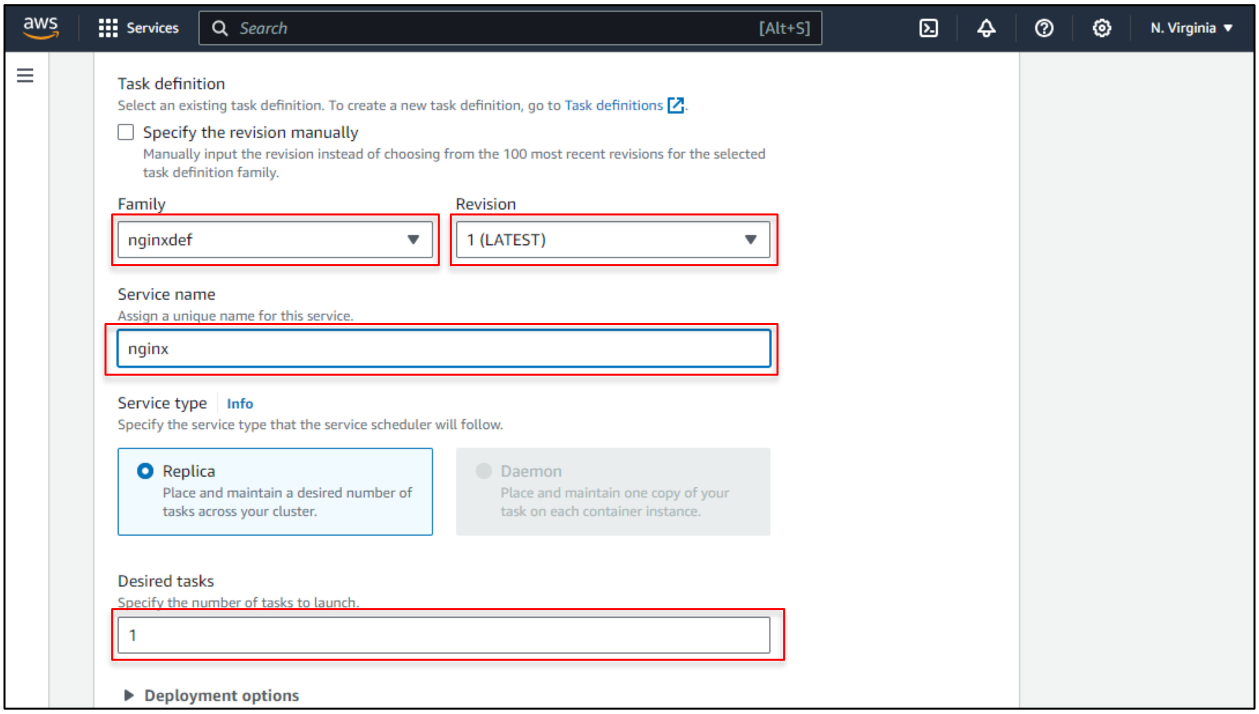
7.2 Click on **Create** under **Services**



7.3 Choose **FARGATE** under Launch Type, select **LATEST** as the platform version, and **Service** as the application type


Capacity provider strategy
Specify a launch strategy to distribute your tasks across one or more capacity providers.
Launch type
Launch tasks directly without the use of a capacity provider strategy.
Launch type Info
Select either managed capacity (Fargate), or custom capacity (EC2 or user-managed, External instances). External instances are registered to your cluster using the ECS Anywhere capability.
FARGATE
Platform version Info
Specify the platform version on which to run your service.
LATEST
Deployment configuration
Application type Info
Specify what type of application you want to run.
Service
Launch a group of tasks handling a long-running computing work that can be stopped and restarted. For example,
Task
Launch a standalone task that runs and terminates. For example, a batch job.

7.4 Choose **nginxdef** as the family (created earlier), **1 (LATEST)** as the revision, **nginx** as the service name, and **1** for the replica count



Task definition

Select an existing task definition. To create a new task definition, go to Task definitions

□ Specify the revision manually

Manually input the revision instead of choosing from the 100 most recent revisions for the selected task definition family.

Family: nginxdef

Revision: 1 (LATEST)

Service name

Assign a unique name for this service.

nginx

Service type Info

Specify the service type that the service scheduler will follow.

● Replica
Place and maintain a desired number of tasks across your cluster.

○ Daemon
Place and maintain one copy of your task on each container instance.

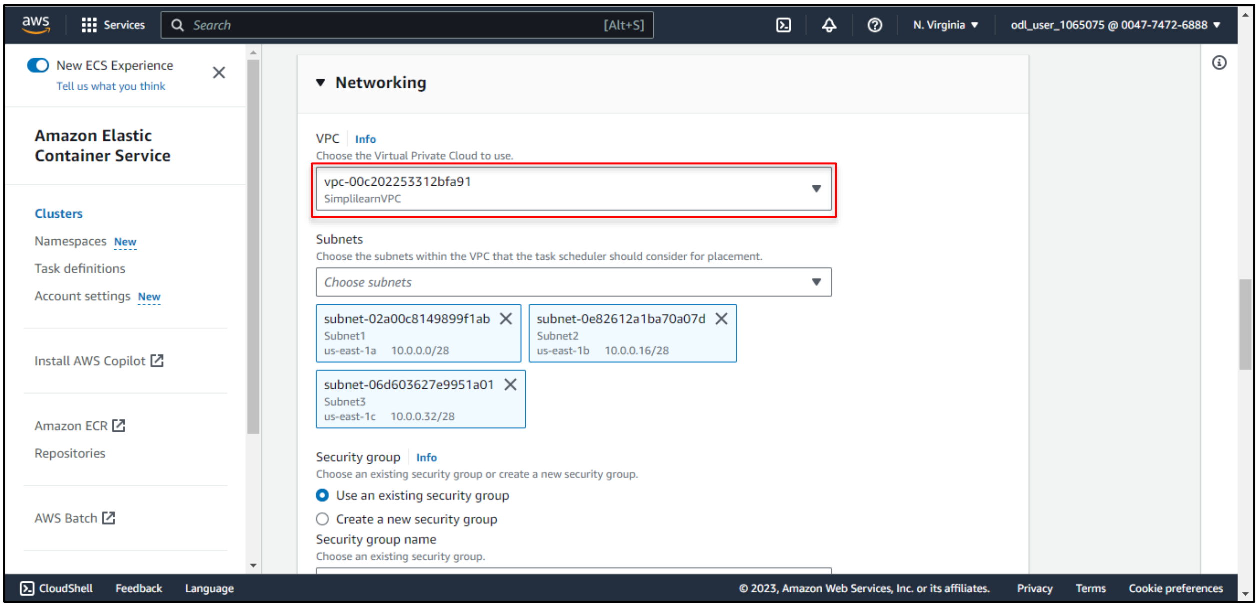
Desired tasks

Specify the number of tasks to launch.

1

▶ Deployment options

7.5 In the Networking section, select the previously created VPC and choose **Application Load Balancer** from the drop-down options for Load Balancing



▼ Networking

VPC Info

Choose the Virtual Private Cloud to use.

vpc-00c202253312bfa91
SimplilearnVPC

Subnets

Choose the subnets within the VPC that the task scheduler should consider for placement.

Choose subnets

subnet-02a00c8149899f1ab ✕
Subnet1
us-east-1a 10.0.0.0/28

subnet-0e82612a1ba70a07d ✕
Subnet2
us-east-1b 10.0.0.16/28

subnet-06d603627e9951a01 ✕
Subnet3
us-east-1c 10.0.0.32/28

Security group Info

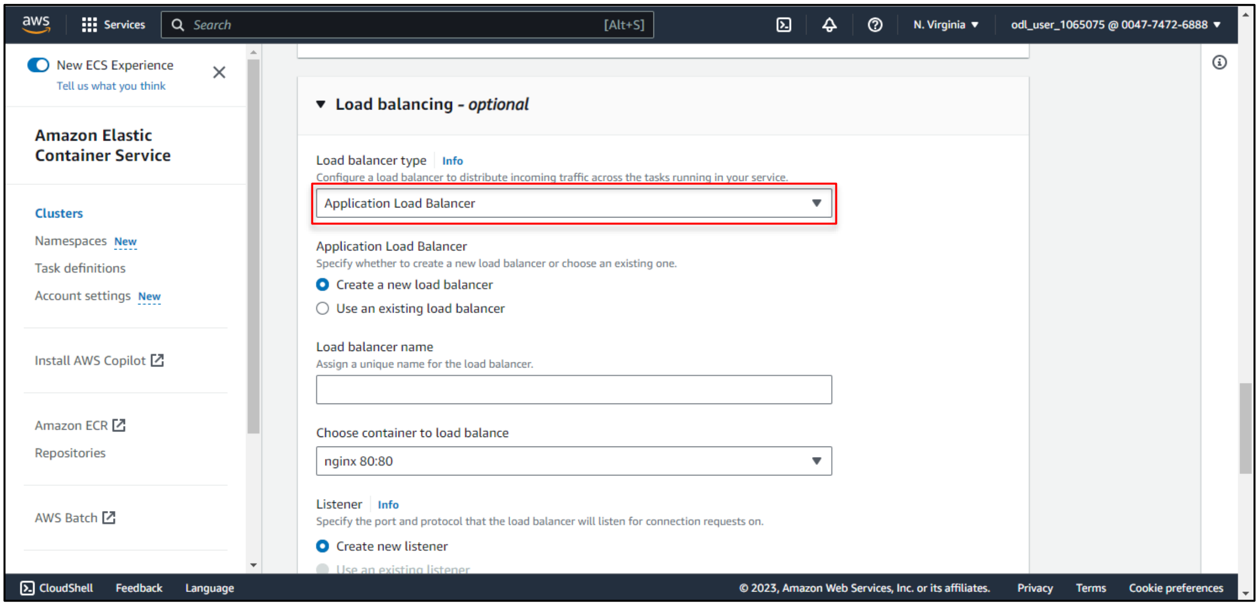
Choose an existing security group or create a new security group.

● Use an existing security group

○ Create a new security group

Security group name

Choose an existing security group.



aws Services Search [Alt+S] N. Virginia ▼ odl_user_1065075 @ 0047-7472-6888 ▼

New ECS Experience
Tell us what you think

Amazon Elastic Container Service

Clusters
Namespaces New
Task definitions
Account settings New
Install AWS Copilot
Amazon ECR
Repositories
AWS Batch

▼ Load balancing - *optional*

Load balancer type Info
Configure a load balancer to distribute incoming traffic across the tasks running in your service.
Application Load Balancer ▼

Application Load Balancer
Specify whether to create a new load balancer or choose an existing one.
● Create a new load balancer
○ Use an existing load balancer

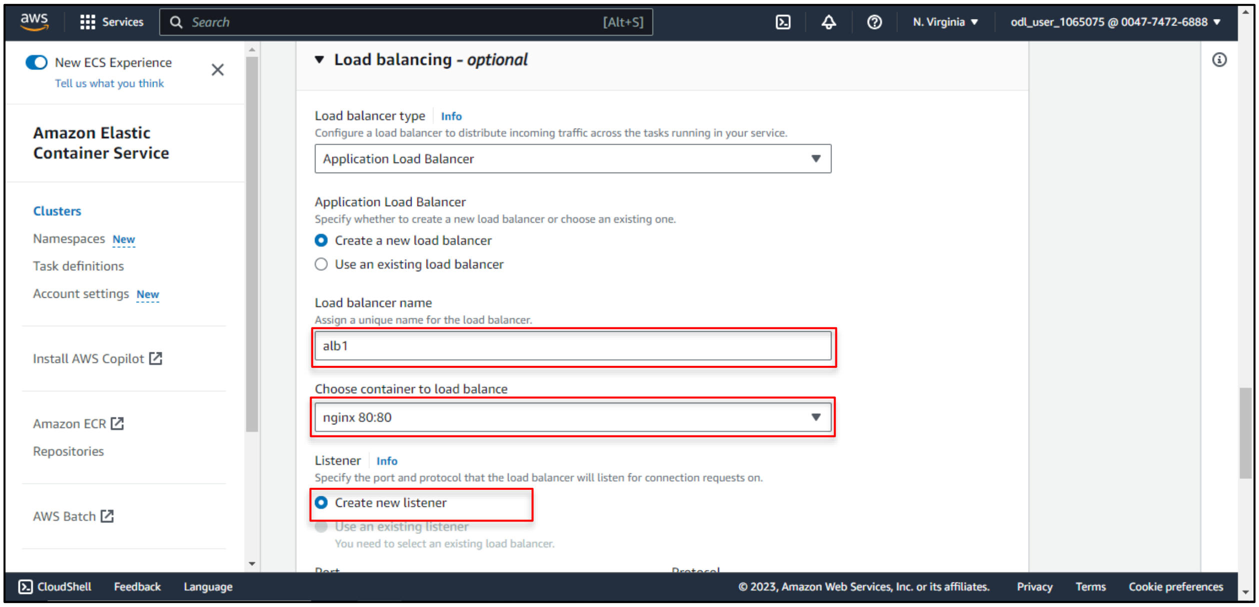
Load balancer name
Assign a unique name for the load balancer.

Choose container to load balance
nginx 80:80 ▼

Listener Info
Specify the port and protocol that the load balancer will listen for connection requests on.
● Create new listener
○ Use an existing listener

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

7.6 Enter **alb1** as the **Load Balancer name**, choose **nginx 80:80** as the container to load balance, select **Create new listener** for the **Listener**, set the port to **80**, choose **create new target group** for the target group, name the target group **target1**, and click **Create**



aws Services Search [Alt+S] N. Virginia ▼ odl_user_1065075 @ 0047-7472-6888 ▼

New ECS Experience
Tell us what you think

Amazon Elastic Container Service

Clusters
Namespaces New
Task definitions
Account settings New
Install AWS Copilot
Amazon ECR
Repositories
AWS Batch

▼ Load balancing - *optional*

Load balancer type Info
Configure a load balancer to distribute incoming traffic across the tasks running in your service.
Application Load Balancer ▼

Application Load Balancer
Specify whether to create a new load balancer or choose an existing one.
● Create a new load balancer
○ Use an existing load balancer

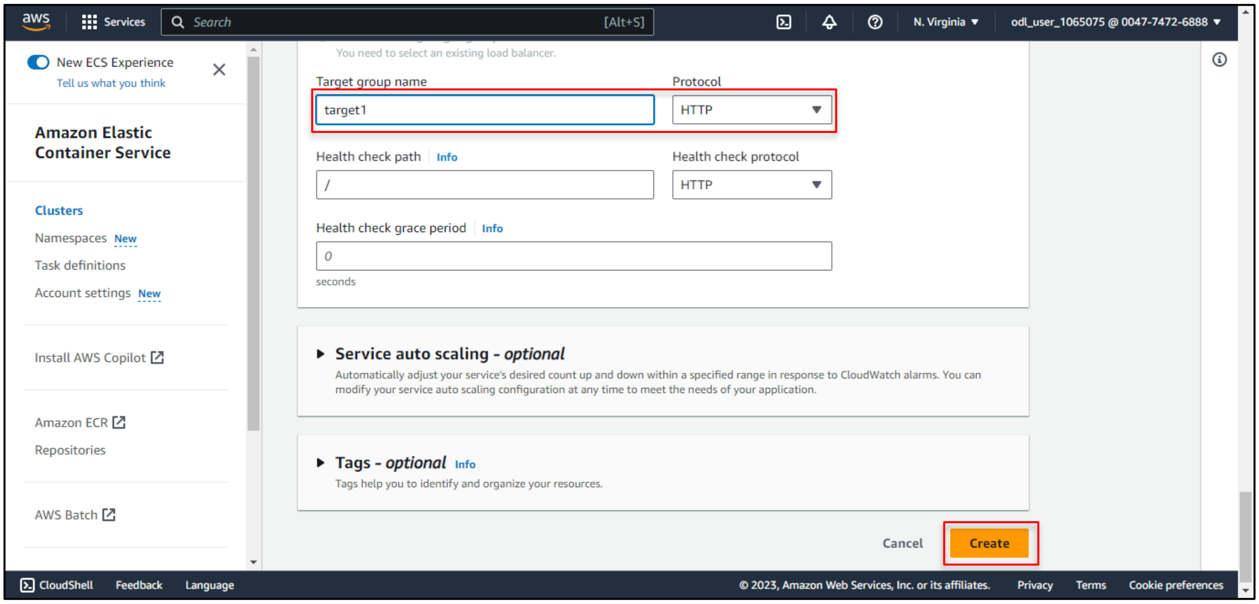
Load balancer name
Assign a unique name for the load balancer.
alb1

Choose container to load balance
nginx 80:80 ▼

Listener Info
Specify the port and protocol that the load balancer will listen for connection requests on.
● Create new listener
○ Use an existing listener
You need to select an existing load balancer.

Port Protocol

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



You need to select an existing load balancer.

Target group name	Protocol
target1	HTTP

Health check path Info	Health check protocol
/	HTTP

Health check grace period Info

0

seconds

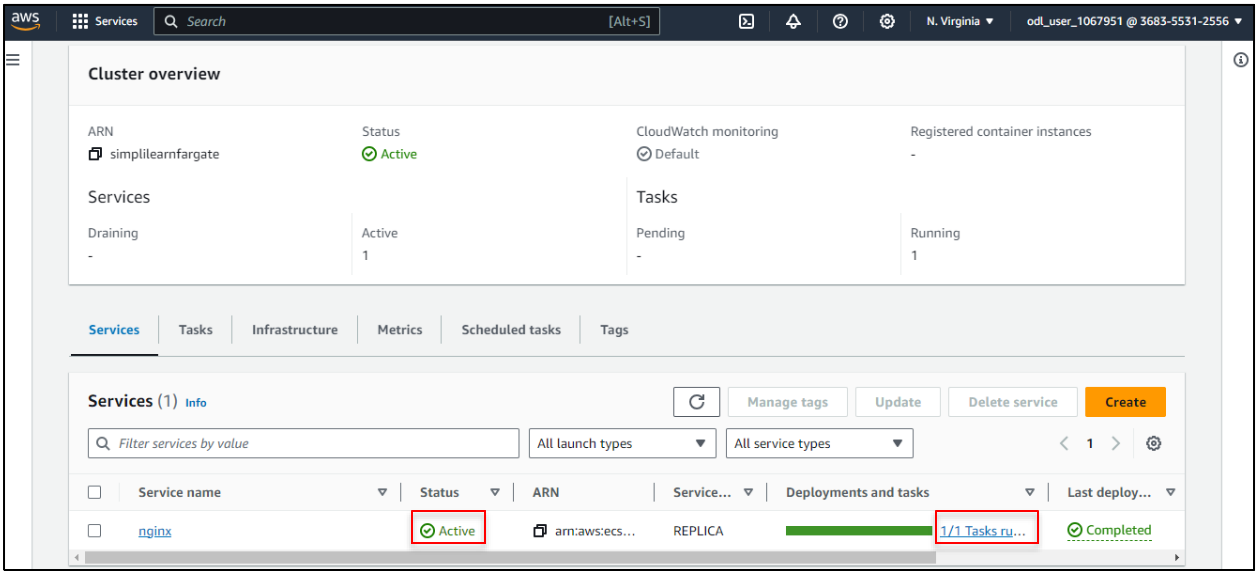
Service auto scaling - *optional*

Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your service auto scaling configuration at any time to meet the needs of your application.

Tags - *optional* Info

Tags help you to identify and organize your resources.

Wait until service creation has been completed and 1/1 tasks are shown as active and running.



Cluster overview

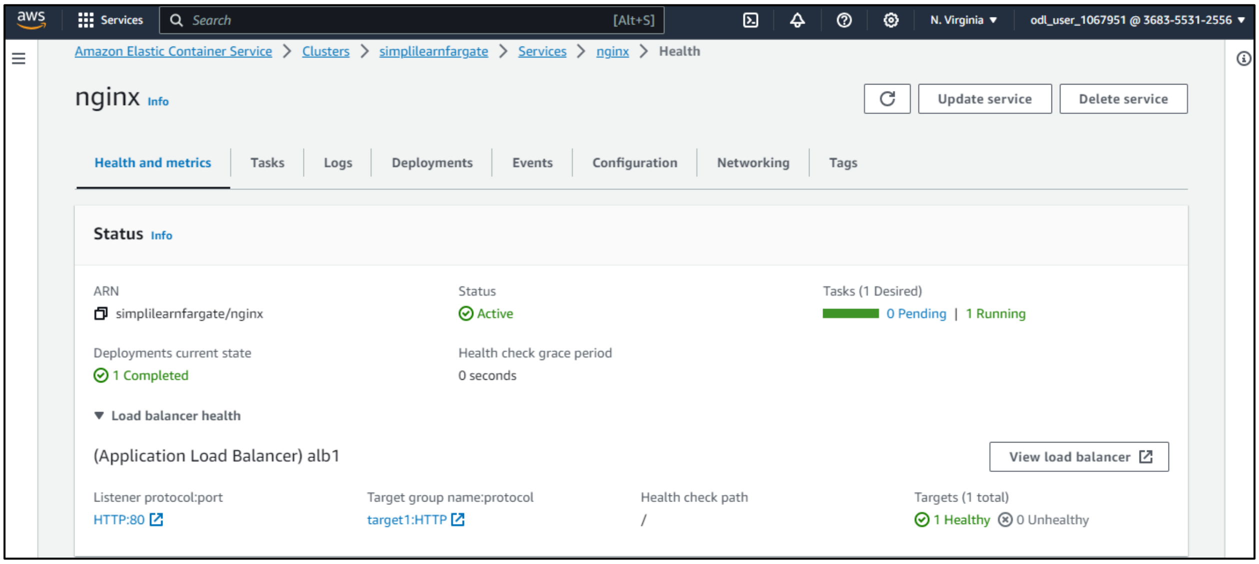
ARN	Status	CloudWatch monitoring	Registered container instances
simplilearnfargate	Active	Default	-

Services		Tasks	
Draining	Active	Pending	Running
-	1	-	1

Services (1) Info

Service name	Status	ARN	Service...	Deployments and tasks	Last deploy...
nginx	Active	arn:aws:ecs...	REPLICA	1/1 Tasks ru...	Completed

7.7 Once the service is running, click on **nginx** to view the service details and **View load balancer** to obtain the URL for accessing the application on the ECS cluster



ARN: simplilearnfargate/nginx

Status: Active

Tasks (1 Desired): 0 Pending | 1 Running

Deployments current state: 1 Completed

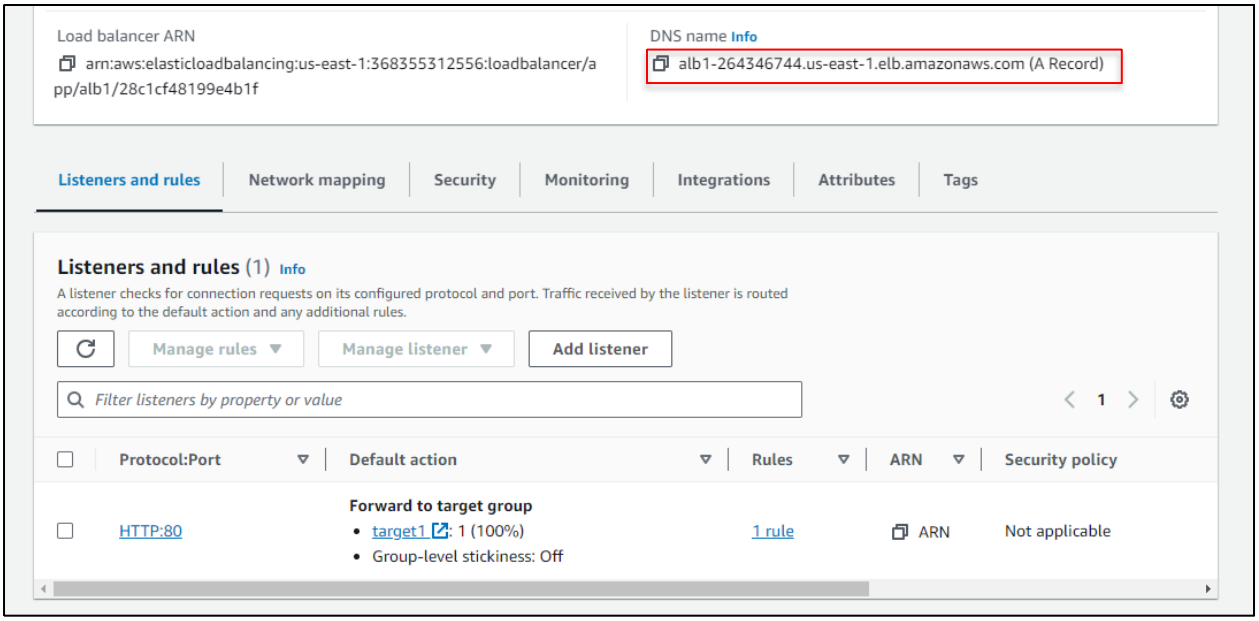
Health check grace period: 0 seconds

▼ Load balancer health

(Application Load Balancer) alb1

Listener protocol:port	Target group name:protocol	Health check path	Targets (1 total)
HTTP:80	target1:HTTP	/	1 Healthy 0 Unhealthy

7.8 Copy the **DNS name** and open it in a new browser



Load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:368355312556:loadbalancer/app/alb1/28c1cf48199e4b1f

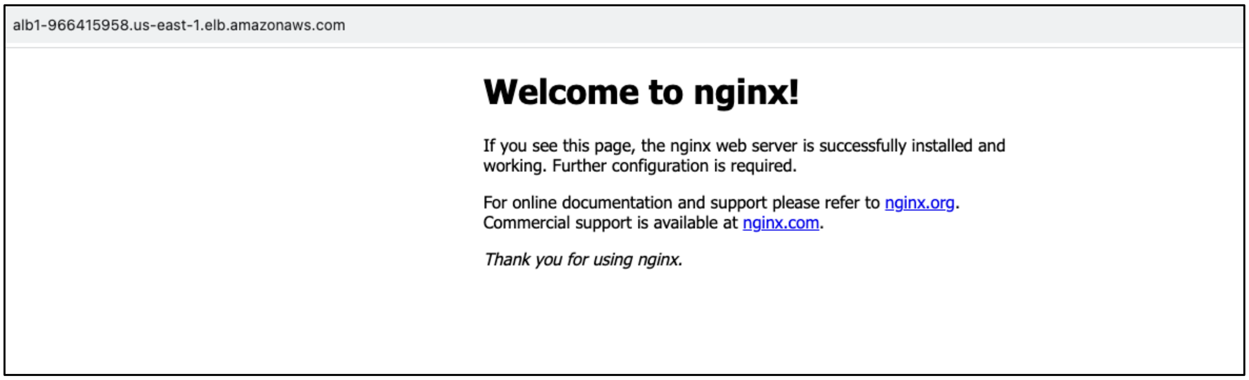
DNS name Info: alb1-264346744.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

□	Protocol:Port	Default action	Rules	ARN	Security policy
□	HTTP:80	Forward to target group • target1: 1 (100%) • Group-level stickiness: Off	1 rule	ARN	Not applicable

7.10 Open the URL in a new browser to see the nginx page load as shown below:



alb1-966415958.us-east-1.elb.amazonaws.com

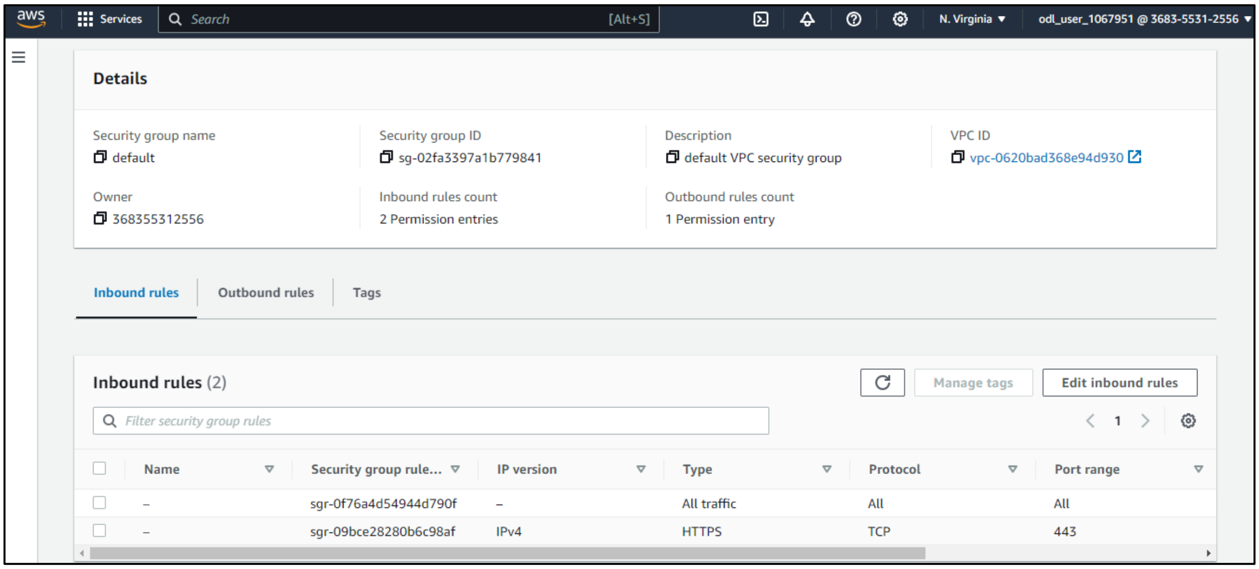
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

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:



Details

Security group name	Security group ID	Description	VPC ID
default	sg-02fa3397a1b779841	default VPC security group	vpc-0620bad368e94d930
Owner	Inbound rules count	Outbound rules count	
368355312556	2 Permission entries	1 Permission entry	

Inbound rules (2)

□	Name	Security group rule...	IP version	Type	Protocol	Port range
□	–	sgr-0f76a4d54944d790f	–	All traffic	All	All
□	–	sgr-09bce28280b6c98af	IPv4	HTTPS	TCP	443

By following these steps, you have successfully integrated dynamic ports with an Application Load Balancer, enhancing the scalability and flexibility of your application's infrastructure. This ensures efficient traffic management and improved service availability.