

## Lesson 10 Demo 03

### Creating an ECS Cluster with Auto Scaling

**Objective:** To demonstrate how to create an ECS cluster with auto scaling and container Insights for enhanced monitoring and automatic adjustment of resource allocation based on demand

**Tools required:** AWS Management Console

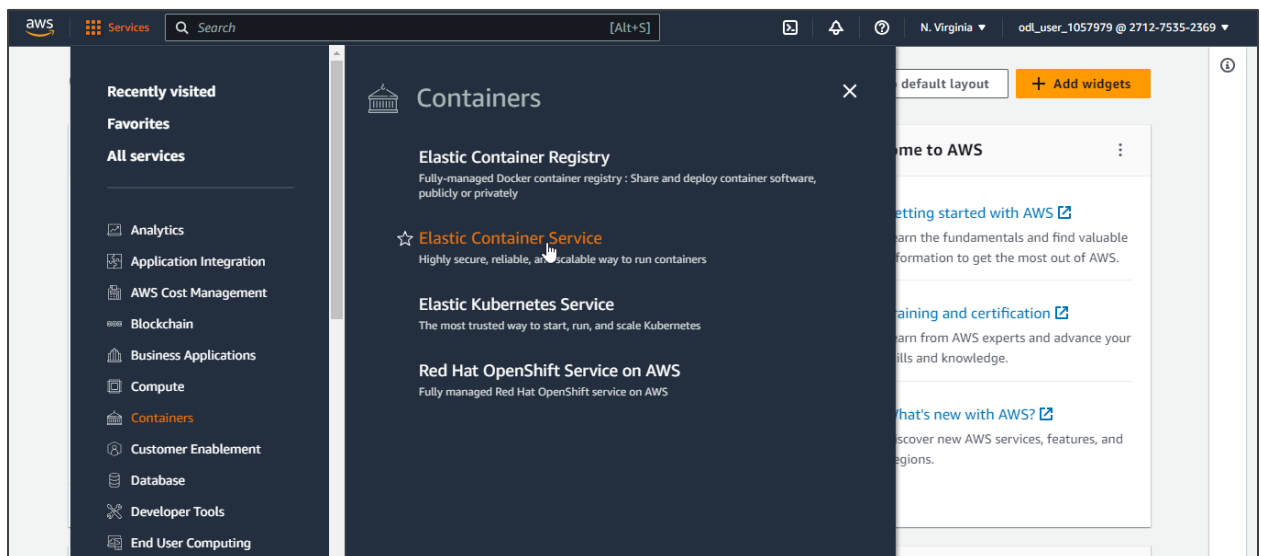
**Prerequisites:** A key pair for an EC2 instance must be created.

Steps to be followed:

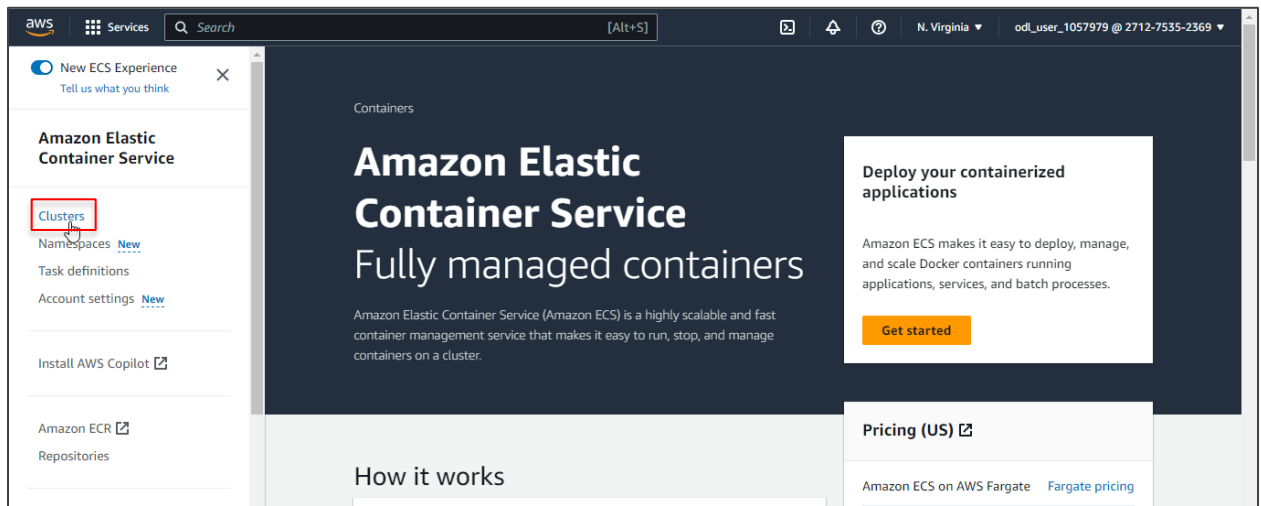
1. Create an ECS cluster
2. Create a task definition
3. Run the task definition on the cluster

#### Step 1: Create an ECS cluster

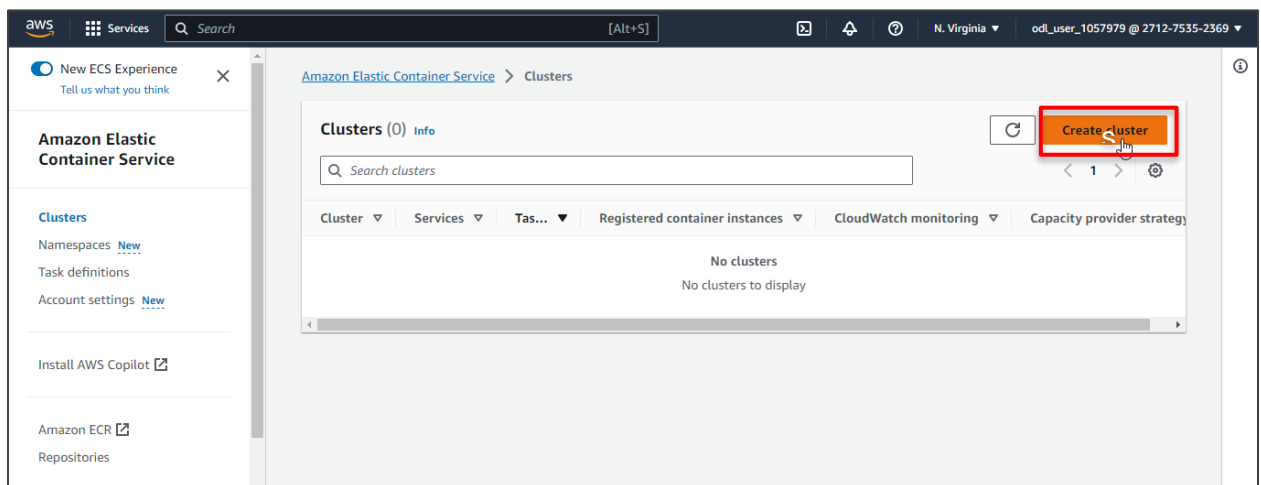
1.1 Navigate to the AWS Console, select **Services** and then choose **ECS** under **Containers**



## 1.2 Click on **Clusters** in the navigation pane



## 1.3 Click **Create cluster**



## 1.4 Create an ECS cluster per the settings shown in the screenshots:

**Amazon Elastic Container Service** > Create cluster

### Create cluster [Info](#)

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

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.

**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.

[Customized](#)

**Amazon Elastic Container Service**

**Clusters**

Namespaces

Task definitions

Account settings

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

Documentation

Discover products

☐ 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.

**Auto Scaling group (ASG)** [Info](#)

Use Auto Scaling groups to scale the Amazon EC2 instances in the cluster.

Create new ASG

**Provisioning model**

Select a provisioning model for your instances

☐ On-demand

With on-demand instances, you pay for compute capacity by the hour, with no long-term commitments or upfront payments.

☒ Spot

Amazon EC2 Spot instances let you take advantage of unused EC2 capacity in the AWS cloud. Spot instances are available at up to a 90% discount compared to on-demand prices.

**Allocation strategy** [Info](#)

Choose the spot instance allocation strategy.

Price capacity optimized (recommended)

**Container instance Amazon Machine Image (AMI)**

Choose the Amazon ECS-optimized AMI for your instance.

**Amazon Elastic Container Service**

- Clusters
- Namespaces
- Task definitions
- Account settings
- Install AWS Copilot
- Amazon ECR
- Repositories
- AWS Batch
- Documentation
- Discover products

**Container instance Amazon Machine Image (AMI)**  
Choose the Amazon ECS-optimized AMI for your instance.  
Amazon Linux 2 (kernel 5.10)

**EC2 instance type**  
Choose based on the workloads you plan to run on this cluster.  
t2.micro (Free tier eligible)  
i386, x86\_64  
1 vCPU 1 GiB Memory

**EC2 instance role**  
An instance role is used by Amazon EC2 instances to make AWS API requests. If you don't already have an instance IAM role created, we can create one for you.  
Create new role

**Desired capacity**  
Specify the number of instances to launch in your cluster.  
Minimum: 2 Maximum: 3

**SSH Key pair**  
If you do not specify a key pair, you can't connect to the instances via SSH unless you choose an AMI that is configured to allow users another way to log in.  
mykey Create a new key pair

## 1.5 Click Create

**Amazon Elastic Container Service**

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

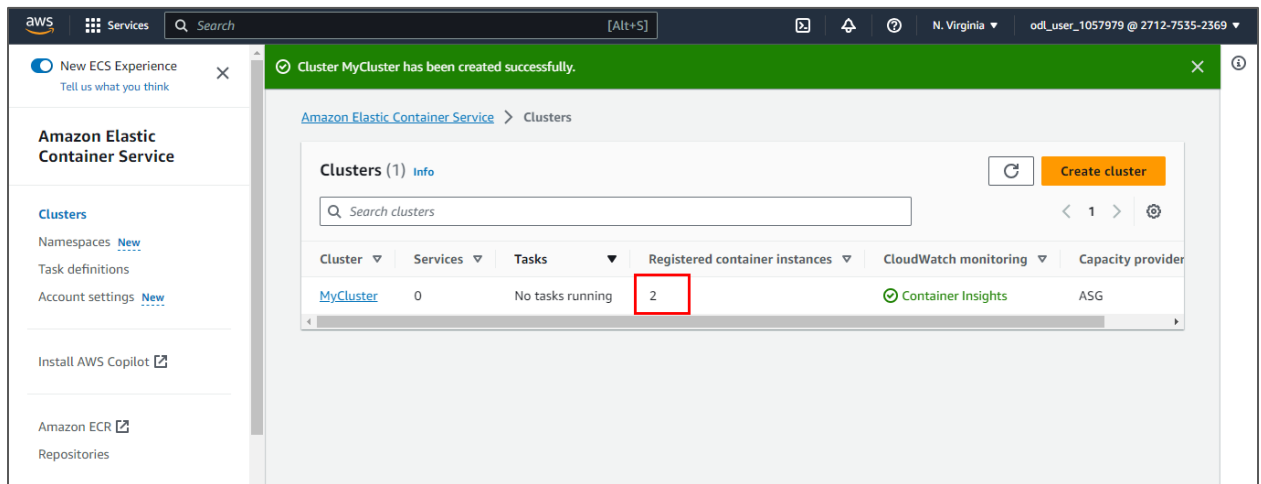
**Monitoring - optional** **Info**  
Container Insights is off by default. When you use Container Insights, there is a cost associated with it.

☒ **Use Container Insights**  
CloudWatch automatically collects metrics for many resources, such as CPU, memory, disk, and network. Container Insights also provides diagnostic information, such as container restart failures, that you use to isolate issues and resolve them quickly. You can also set CloudWatch alarms on metrics that Container Insights collects.

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

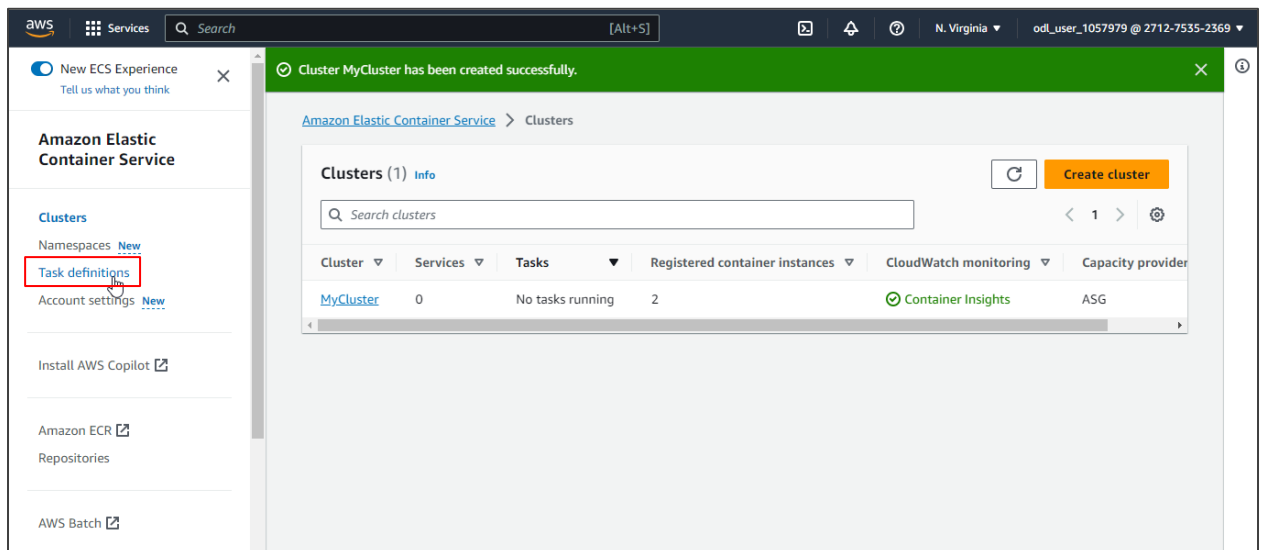
Cancel **Create**

**Note:** Please wait a few minutes for the EC2 instances to register with the cluster.

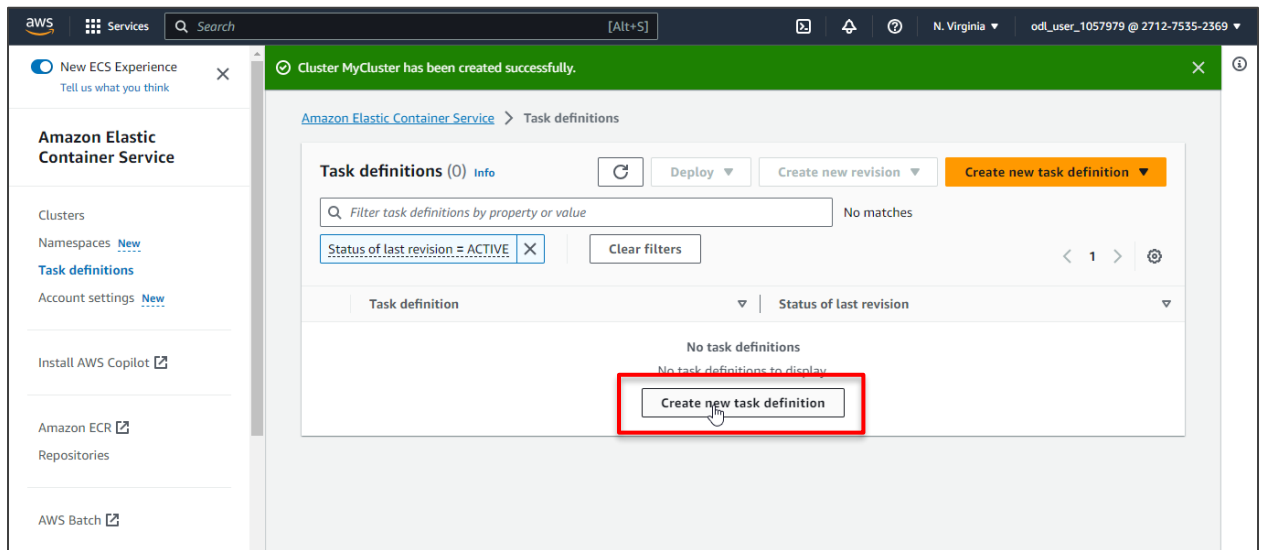


## Step 2: Create a task definition

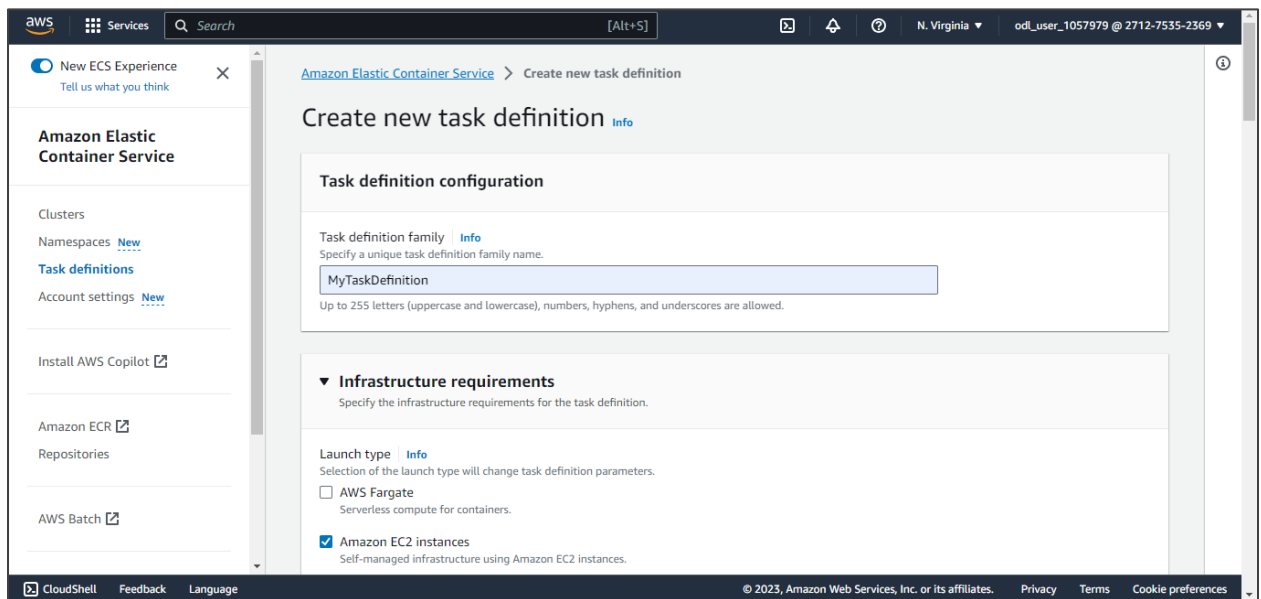
### 2.1 In the navigation panel, click Task definitions



## 2.2 Click Create new task definition



## 2.3 Create the task definition per the settings shown in the screenshots:



**Container - 1** [Info](#) Essential container Remove

**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:

**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	
<input type="text" value="80"/>	<input type="text" value="TCP"/>	<input type="text" value="container-80-tcp"/>	<input type="text" value="HTTP"/>	<span>Remove</span>

Add more port mappings

**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

## 2.4 Click Create

**Storage - optional**

**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.

Add volume

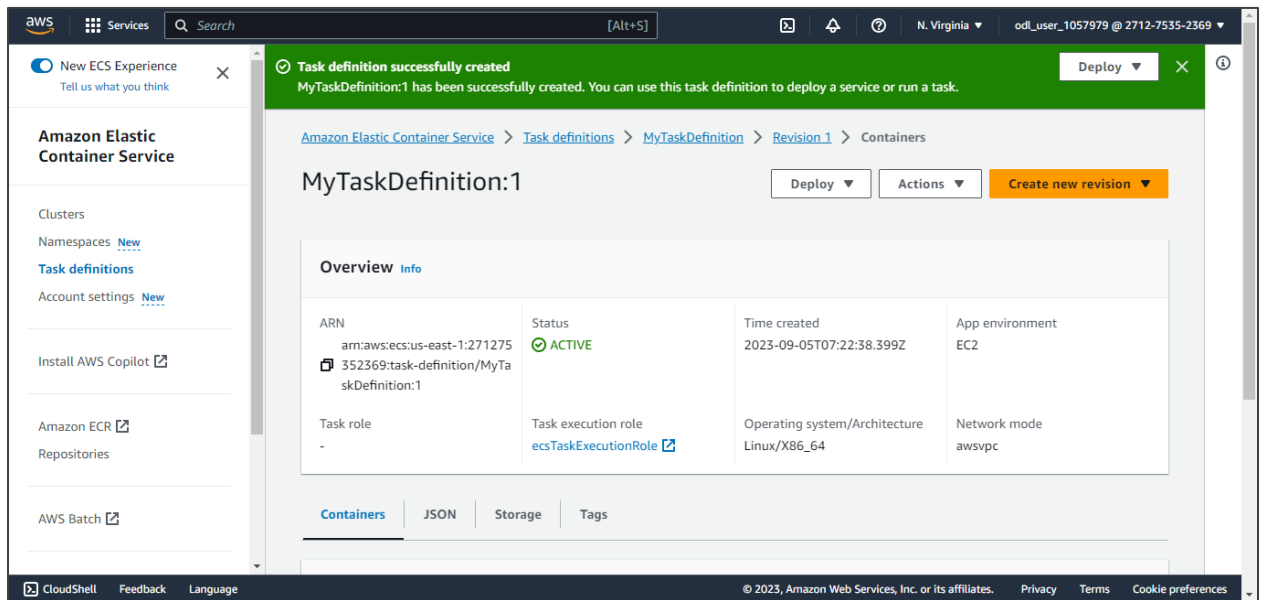
**Volumes from** [Info](#)  
Mount data volumes from another container.

Add volume from

**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.

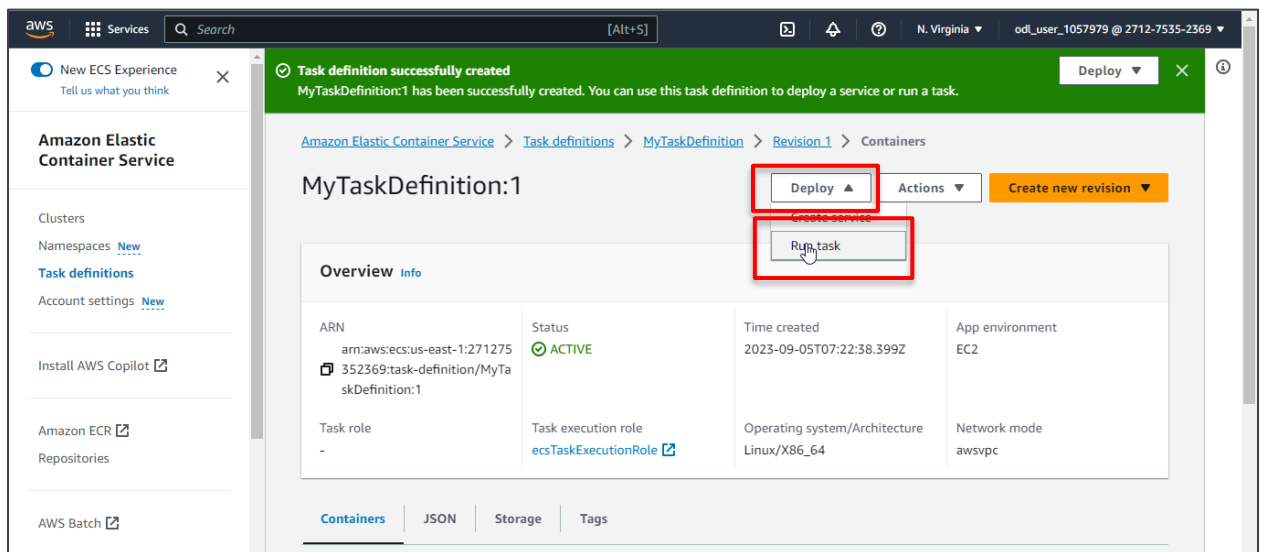
Cancel Create



The task definition has been created successfully.

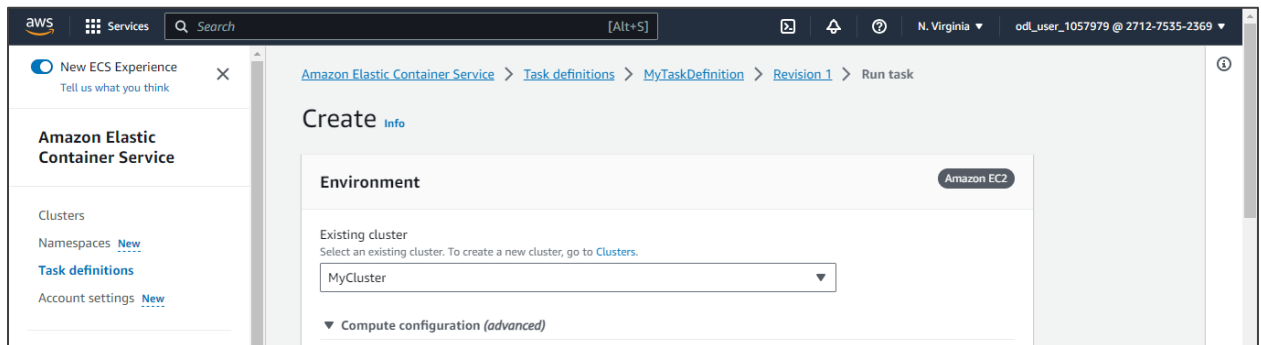
## Step 3: Run the task definition on the cluster

### 3.1 Select **Deploy** and click **Run task** to run the task definition

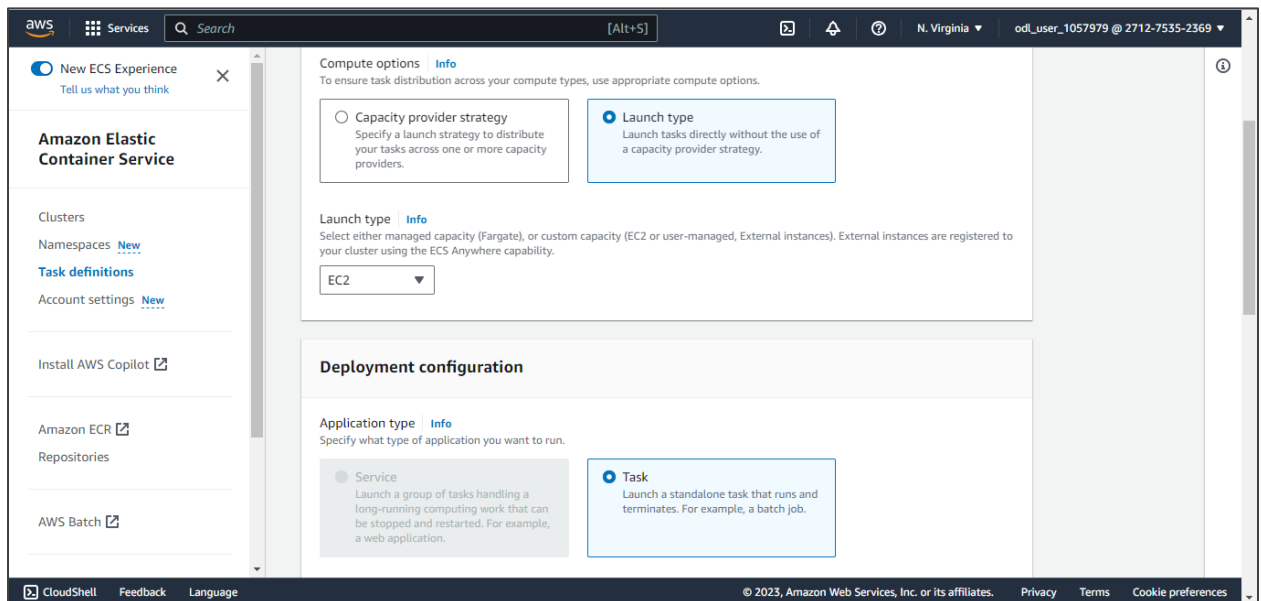




### 3.2 Choose the cluster that you created in the previous step



### 3.3 Configure the other settings per the screenshots:



**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: MyTaskDefinition  
Revision: 1 (LATEST)

**Desired tasks**  
Specify the number of tasks to launch.  
1

**Task group**  
All tasks with the same task group name are considered as a set when performing spread placement.

► **Networking**

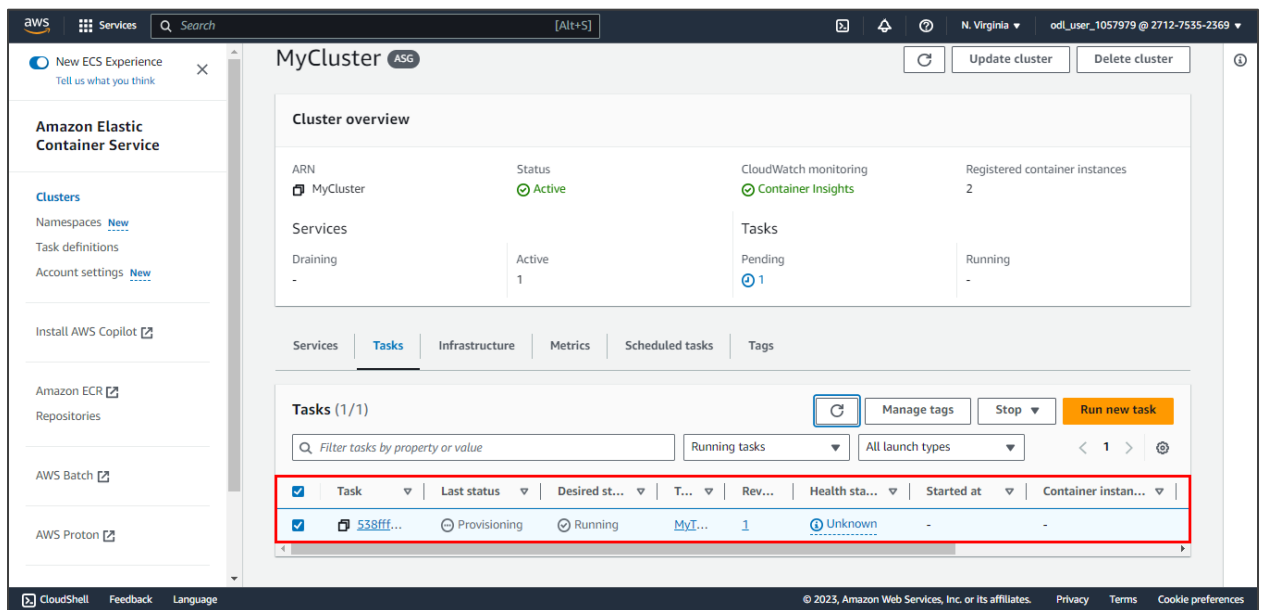
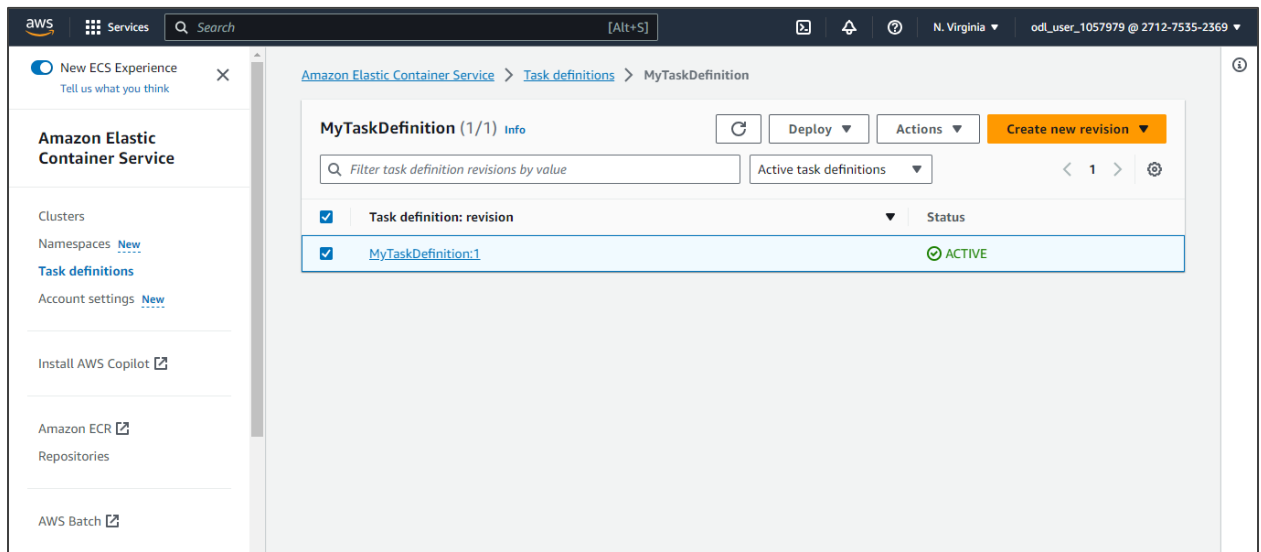
► **Task Placement** [Info](#)  
Lets you customize how tasks are placed on instances within your cluster. Different placement strategies are available to optimize for availability and efficiency.

### 3.4 Choose Create

Cancel **Create**

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

The task definition has been successfully executed.



A task has been added to **MyCluster** and triggered successfully as shown in the screenshot.

By following these steps, you have successfully created an ECS Cluster with Auto Scaling, defined and executed tasks within it.