I. CREATE CUSTOM IMAGE

Install Docker on AMAZON LINUX INSTANCE.

1. Launch an instance with the Amazon Linux 2 AMI.

2. Connect to instance.

3. Update the installed packages and package cache on your instance.

```
sudo yum update -y
```

4. Install the docker Engine

```
sudo yum install docker
```

5.Start the Docker service.

```
sudo service docker start
```

6. View the status of Docker

```
sudo service docker status
```

7. Add the ec2-user to the docker group so you can execute Docker commands without using sudo.

```
sudo usermod -a -G docker ec2-user
```

8. Verify that you can run Docker commands without sudo.

```
docker info
```

II. Create a Docker image

1. Create a file called Dockerfile. A Dockerfile is a manifest that describes the base image to use for your Docker image and what you want installed and running on it.

```
touch Dockerfile
```

2. Edit the Dockerfile you just created and add the following content.

```
Vi Dockerfile
```

```
FROM ubuntu:18.04

# Install dependencies

RUN apt-get update && \

 apt-get -y install apache2
```

```
# Install apache and write hello world message

RUN echo 'Hello World!' > /var/www/html/index.html

# Configure apache

RUN echo '. /etc/apache2/envvars' > /root/run_apache.sh && \

 echo 'mkdir -p /var/run/apache2' >> /root/run_apache.sh && \

 echo 'mkdir -p /var/lock/apache2' >> /root/run_apache.sh && \

 echo '/usr/sbin/apache2 -D FOREGROUND' >> /root/run_apache.sh && \

 chmod 755 /root/run_apache.sh

EXPOSE 80

CMD /root/run_apache.sh
```

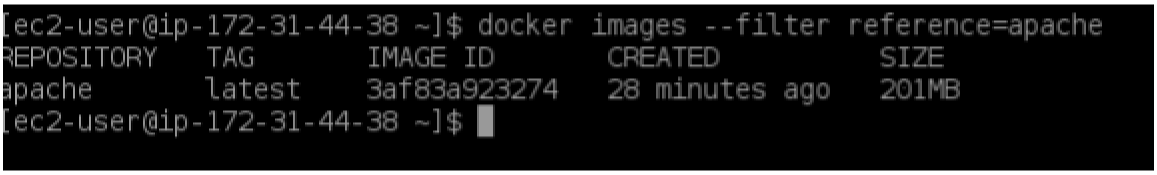
This Dockerfile uses the Ubuntu 18.04 image. The RUN instructions update the package caches, install some software packages for the web server, and then write the "Hello World!" content to the web server's document root. The EXPOSE instruction exposes port 80 on the container, and the CMD instruction starts the web server.

3. Build the Docker image from your Dockerfile.

```
docker build -t apache .
```

4.Run docker images to verify that the image was created correctly.

```
docker images --filter reference=apache
```



```
[ec2-user@ip-172-31-44-38 ~]$ docker images --filter reference=apache
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
apache        latest    3af83a923274   28 minutes ago   201MB
[ec2-user@ip-172-31-44-38 ~]$ █
```

5. Open a browser and point to the server that is running Docker and hosting your container.

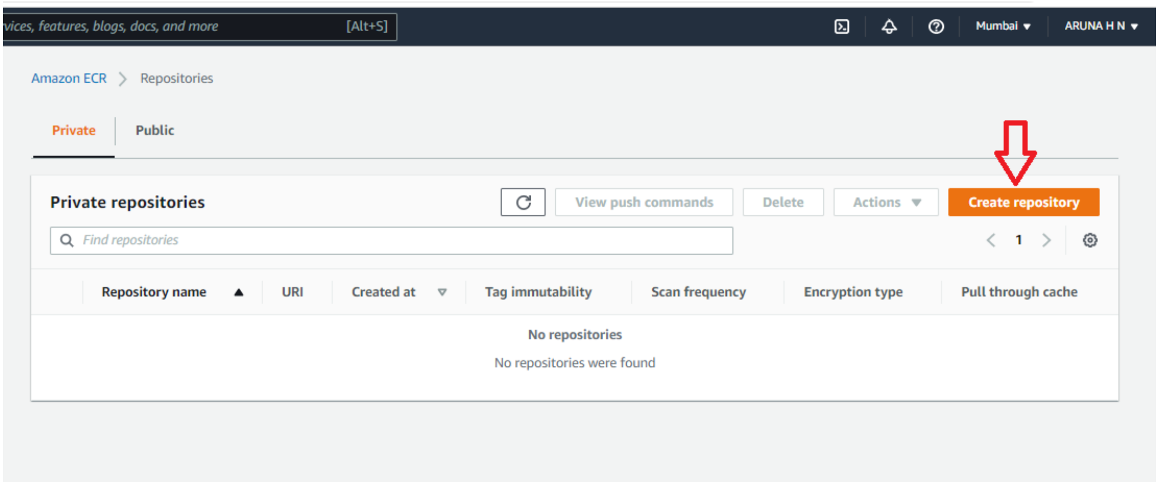
If you are using an EC2 instance, this is the Public DNS value for the server, which is the same address you use to connect to the instance with SSH. Make sure that the security group for your instance allows inbound traffic on port 80.

If you are running Docker locally, point your browser to http://localhost/.

6. Stop the Docker container by typing Ctrl + c.

Push your image to Amazon Elastic Container Registry

1. Goto Amazon ECR >> Repository >>



Private repositories

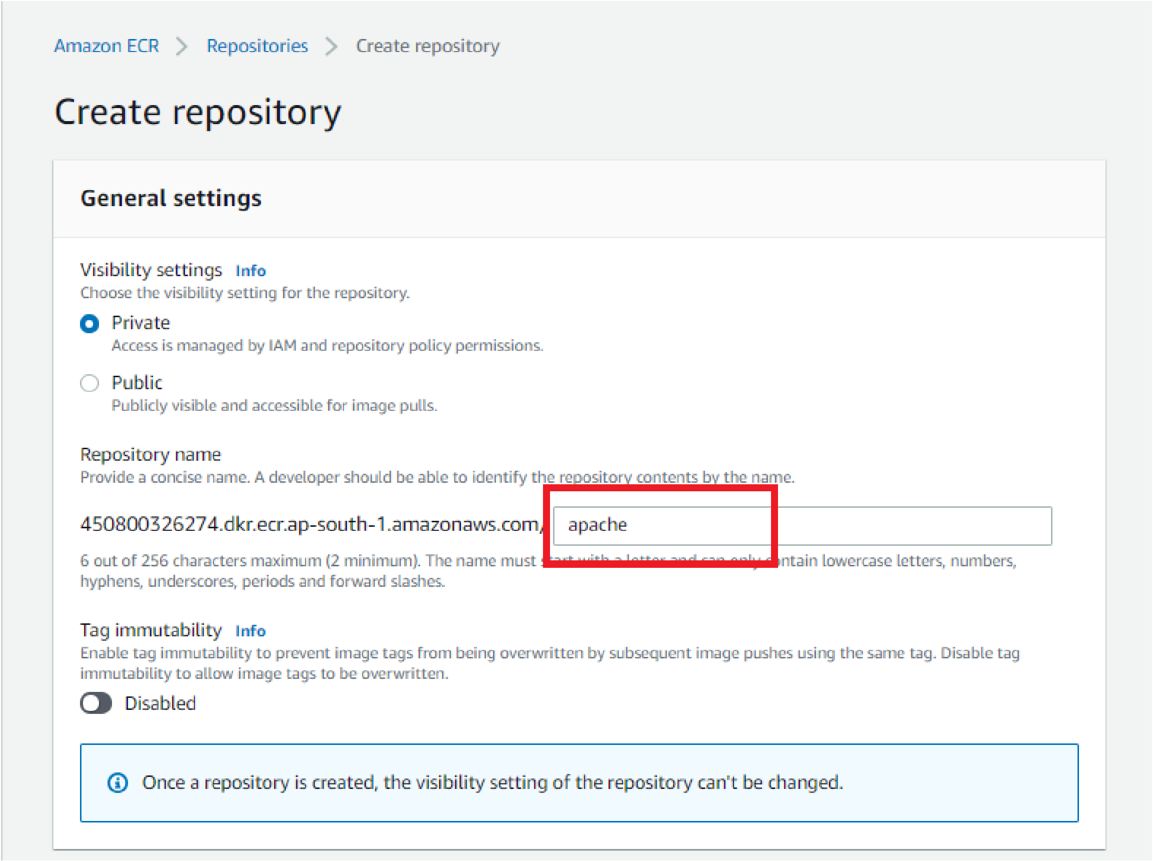
Find repositories

Repository name ▲	URI	Created at ▽	Tag immutability	Scan frequency	Encryption type	Pull through cache

No repositories

No repositories were found

2. Give some Repository name



Create repository

General settings

Visibility settings Info
Choose the visibility setting for the repository.

● Private
Access is managed by IAM and repository policy permissions.

○ Public
Publicly visible and accessible for image pulls.

Repository name
Provide a concise name. A developer should be able to identify the repository contents by the name.

450800326274.dkr.ecr.ap-south-1.amazonaws.com/ apache

6 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

Tag immutability Info
Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.

Disabled

ⓘ Once a repository is created, the visibility setting of the repository can't be changed.

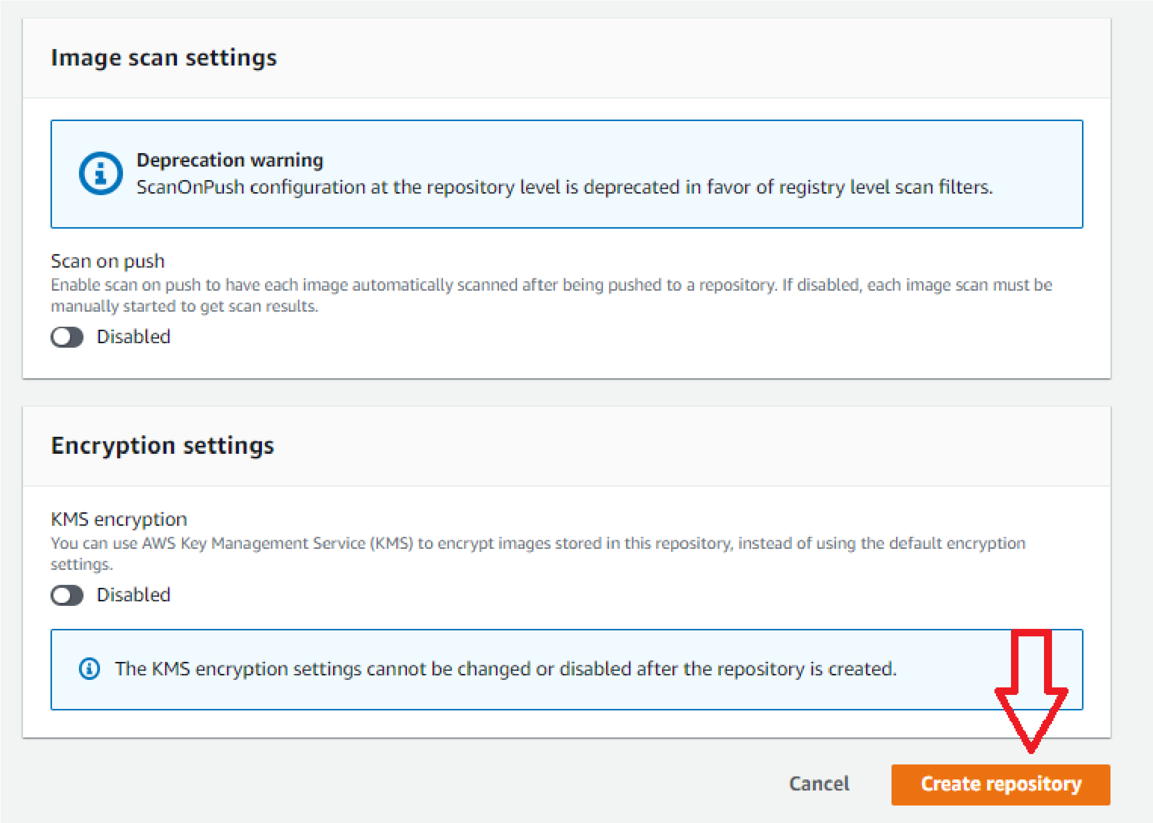


Image scan settings

Deprecation warning
ScanOnPush configuration at the repository level is deprecated in favor of registry level scan filters.

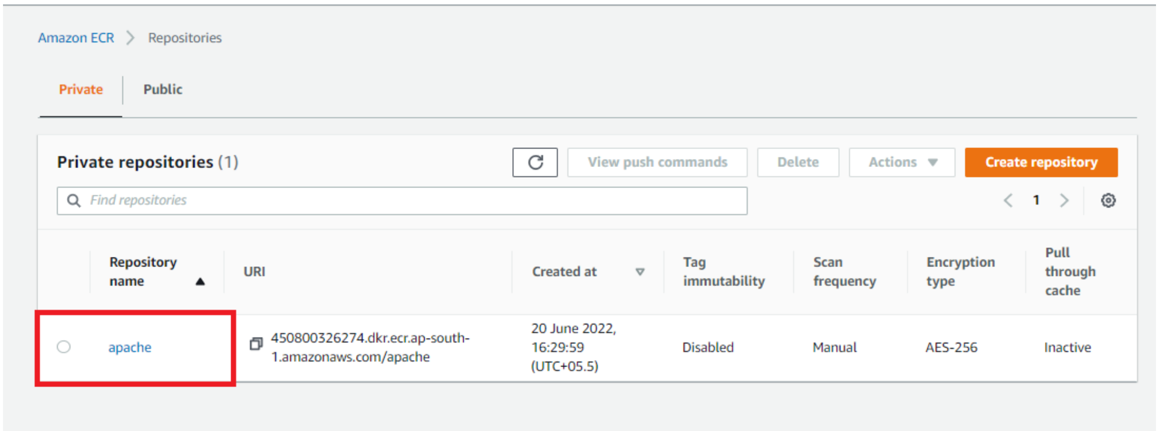
Scan on push
Enable scan on push to have each image automatically scanned after being pushed to a repository. If disabled, each image scan must be manually started to get scan results.
Disabled

Encryption settings

KMS encryption
You can use AWS Key Management Service (KMS) to encrypt images stored in this repository, instead of using the default encryption settings.
Disabled

The KMS encryption settings cannot be changed or disabled after the repository is created.

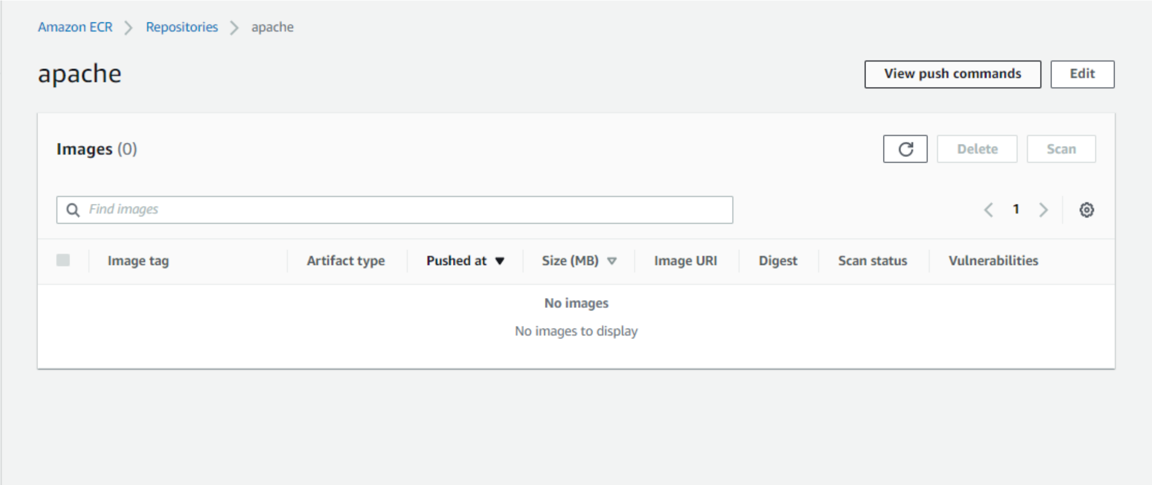
3. Click on Repository name



Private repositories (1)

	Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type	Pull through cache
○	apache	450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache	20 June 2022, 16:29:59 (UTC+05.5)	Disabled	Manual	AES-256	Inactive

4. Here no images are there . so we need to push docker images from our local m/c. so click View push commands.



Amazon ECR > Repositories > apache

apache

View push commands | Edit

Images (0)

Delete | Scan

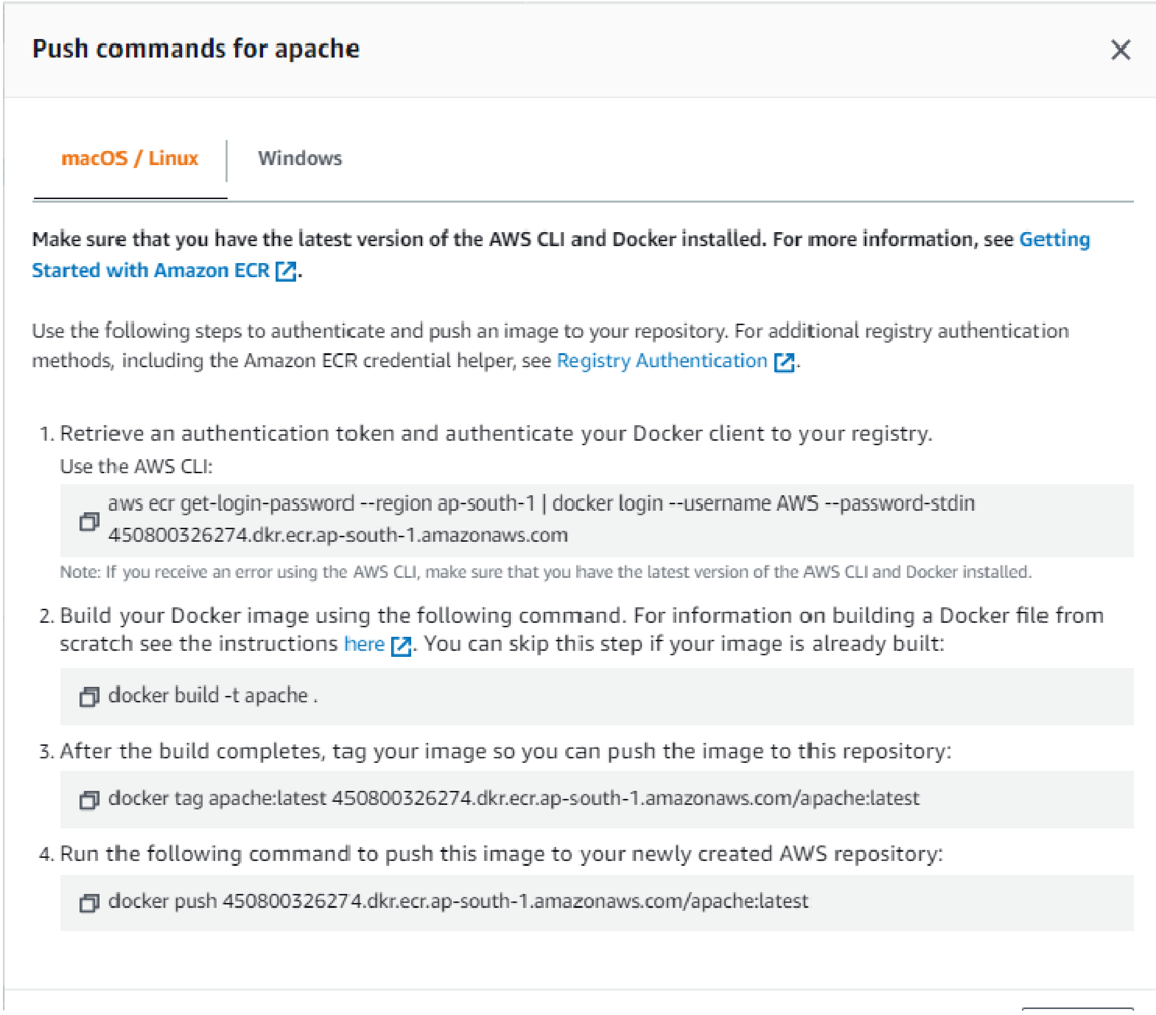
Find images

□	Image tag	Artifact type	Pushed at ▼	Size (MB) ▽	Image URI	Digest	Scan status	Vulnerabilities

No images

No images to display

5. Run the commands one by one on the local m/c where docker file is present.



Push commands for apache

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see Getting Started with Amazon ECR.

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see Registry Authentication.

1. Retrieve an authentication token and authenticate your Docker client to your registry.
   Use the AWS CLI:

```
aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 450800326274.dkr.ecr.ap-south-1.amazonaws.com
```

   Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.

2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions here. You can skip this step if your image is already built:

```
docker build -t apache .
```

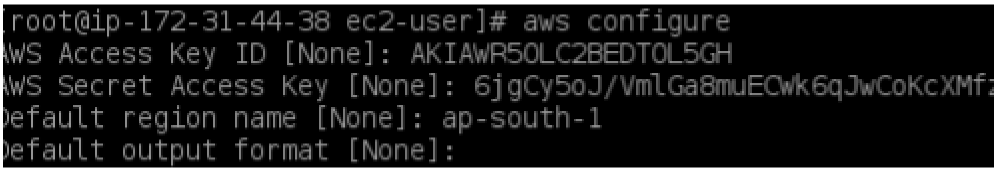
3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag apache:latest 450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache:latest
```

4. Run the following command to push this image to your newly created AWS repository:

```
docker push 450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache:latest
```

6. since the first code uses aws commands we need to configure aws in CLI.

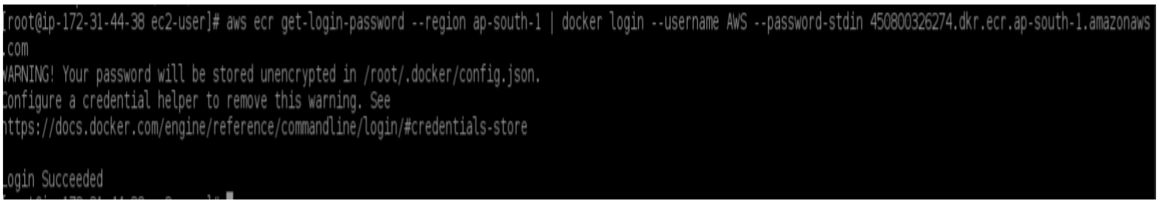


```
[root@ip-172-31-44-38 ec2-user]# aws configure
WS Access Key ID [None]: AKIAWR5OLC2BEDTOL5GH
WS Secret Access Key [None]: 6jgCy5oJ/VmlGa8muECWk6qJwCoKcXMf
efault region name [None]: ap-south-1
efault output format [None]:
```

7. Run this Command to Retrieve an authentication token and authenticate your Docker client to your registry.

Use the AWS CLI:

aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 450800326274.dkr.ecr.ap-south-1.amazonaws.com



```
[root@ip-172-31-44-38 ec2-user]# aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 450800326274.dkr.ecr.ap-south-1.amazonaws
.com
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

8. Build Docker image using the following command

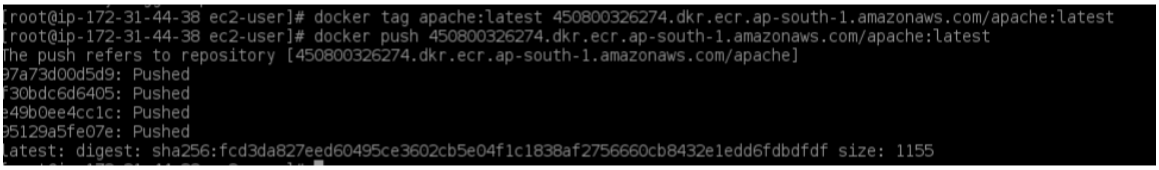
docker build -t apache .

9. After the build completes, tag your image so you can push the image to this repository:

docker tag apache:latest 450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache:latest

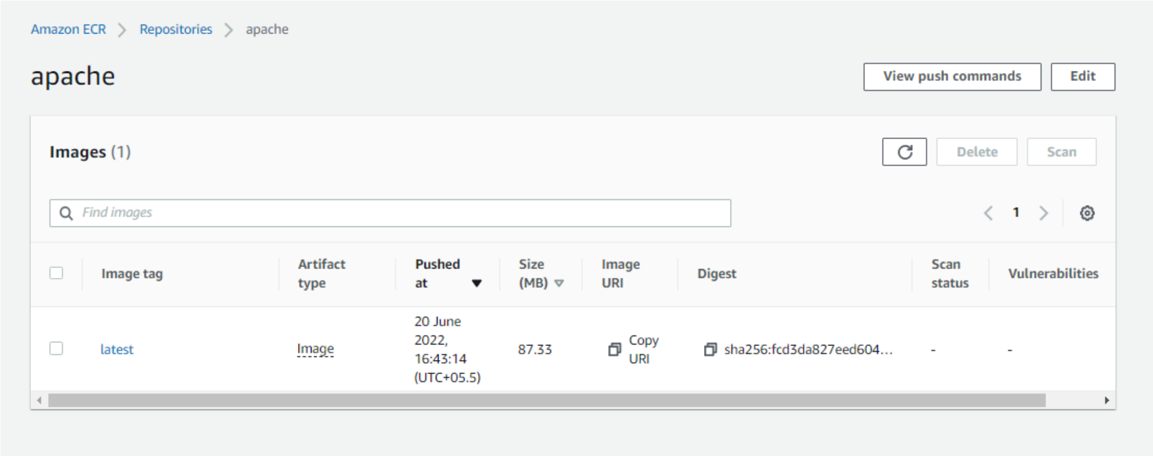
10. Run the following command to push this image to your newly created AWS repository:

docker push 450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache:latest



```
[root@ip-172-31-44-38 ec2-user]# docker tag apache:latest 450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache:latest
[root@ip-172-31-44-38 ec2-user]# docker push 450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache:latest
The push refers to repository [450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache]
97a73d00d5d9: Pushed
f30bdc6d6405: Pushed
e49b0ee4cc1c: Pushed
95129a5fe07e: Pushed
latest: digest: sha256:fcd3da827eed60495ce3602cb5e04f1c1838af2756660cb8432e1edd6fdbdfdf size: 1155
```

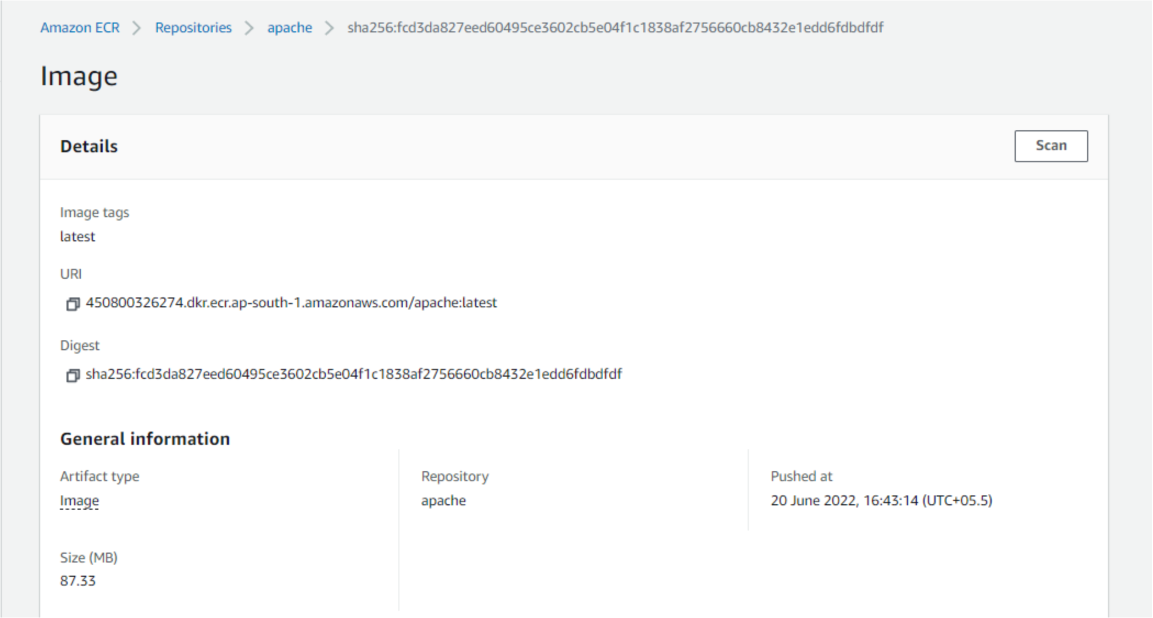
11. Now the Image Pushed into repository



Images (1)

□	Image tag	Artifact type	Pushed at ▼	Size (MB) ▽	Image URI	Digest	Scan status	Vulnerabilities
□	latest	Image	20 June 2022, 16:43:14 (UTC+05.5)	87.33	Copy URI	sha256:fcd3da827eed604...	-	-

12. Detils of the Image



Amazon ECR > Repositories > apache > sha256:fcd3da827eed60495ce3602cb5e04f1c1838af2756660cb8432e1edd6fdbdfdf

Image

Details

Image tags
latest

URI
450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache:latest

Digest
sha256:fcd3da827eed60495ce3602cb5e04f1c1838af2756660cb8432e1edd6fdbdfdf

General information

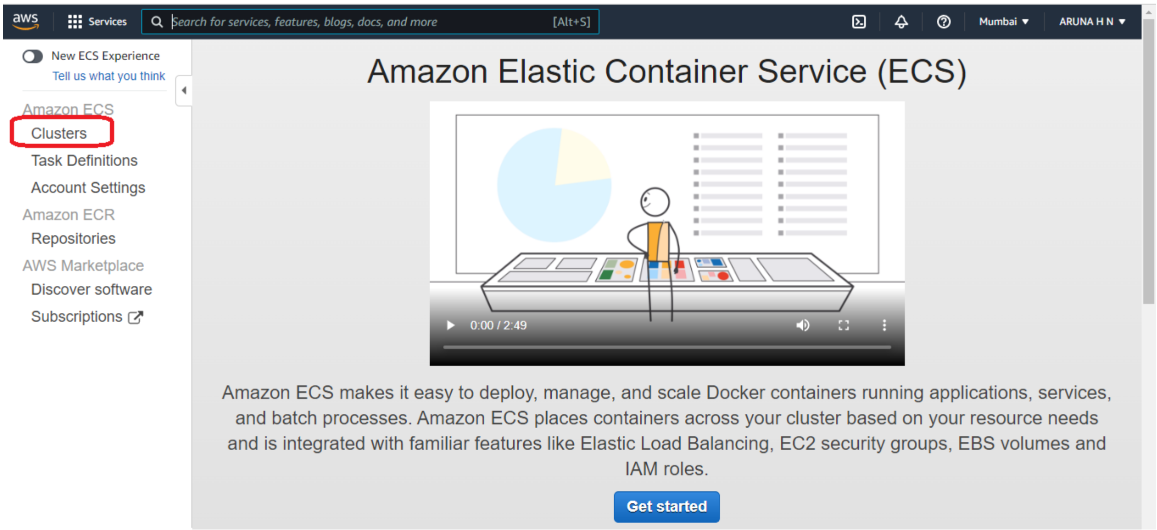
Artifact type	Repository	Pushed at
Image	apache	20 June 2022, 16:43:14 (UTC+05.5)

Size (MB)
87.33

To create Cluster

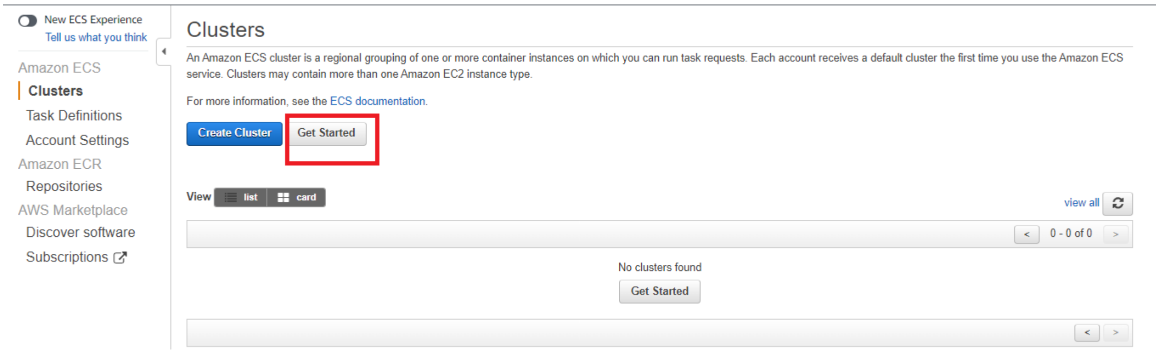
1. Goto ECS in AWS Management Console

2. Select Clusters for creating Cluster



Amazon Elastic Container Service (ECS)

Amazon ECS makes it easy to deploy, manage, and scale Docker containers running applications, services, and batch processes. Amazon ECS places containers across your cluster based on your resource needs and is integrated with familiar features like Elastic Load Balancing, EC2 security groups, EBS volumes and IAM roles.



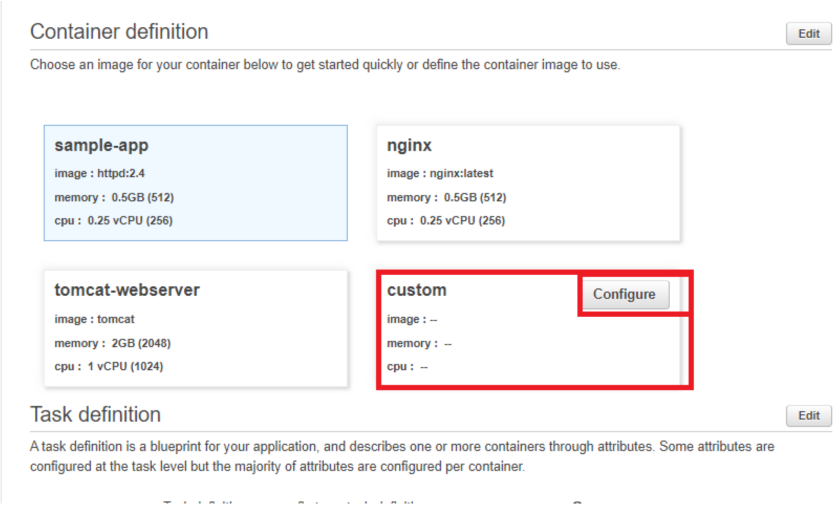
Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

For more information, see the ECS documentation.

No clusters found

3. In Container Defination select Custom Configure



Container definition

Choose an image for your container below to get started quickly or define the container image to use.

sample-app
image : httpd:2.4
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

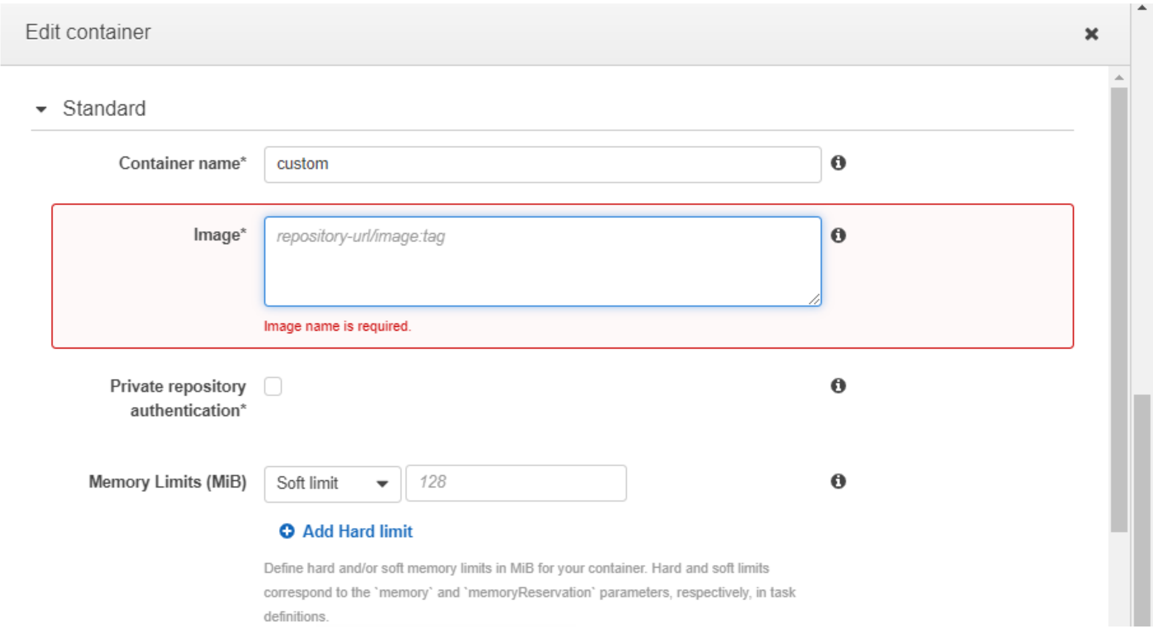
nginx
image : nginx:latest
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

tomcat-webserver
image : tomcat
memory : 2GB (2048)
cpu : 1 vCPU (1024)

custom
image : --
memory : --
cpu : --

Task definition

A task definition is a blueprint for your application, and describes one or more containers through attributes. Some attributes are configured at the task level but the majority of attributes are configured per container.



Edit container

Container name* custom

Image* repository-url/image:tag

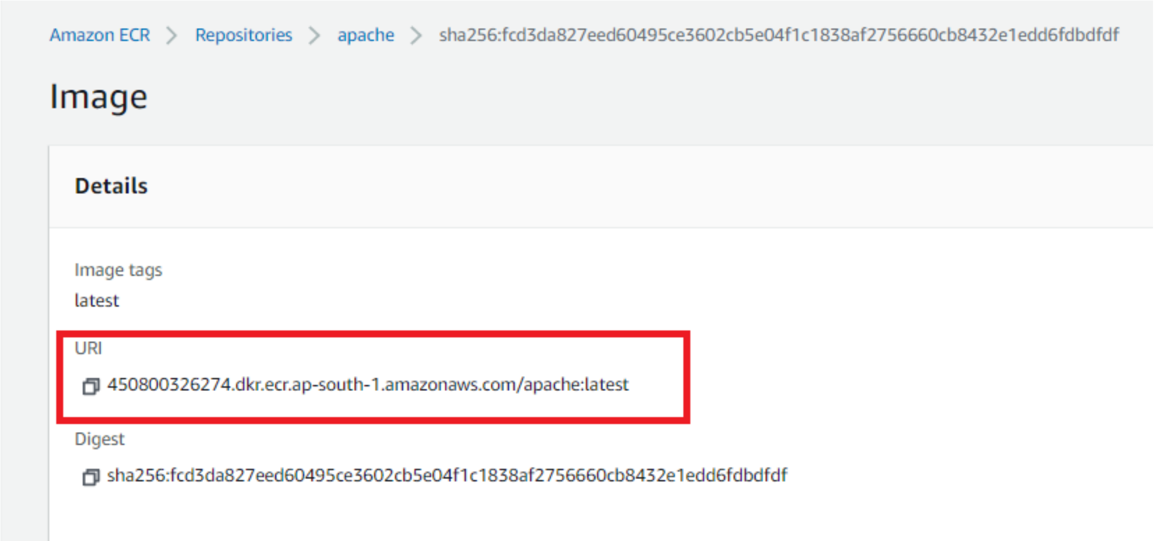
Image name is required.

Private repository authentication* □

Memory Limits (MiB) Soft limit 128

Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, in task definitions.

paste the Image URI Here for example 450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache:latest



Amazon ECR > Repositories > apache > sha256:fcd3da827eed60495ce3602cb5e04f1c1838af2756660cb8432e1edd6fdbdfdf

Image

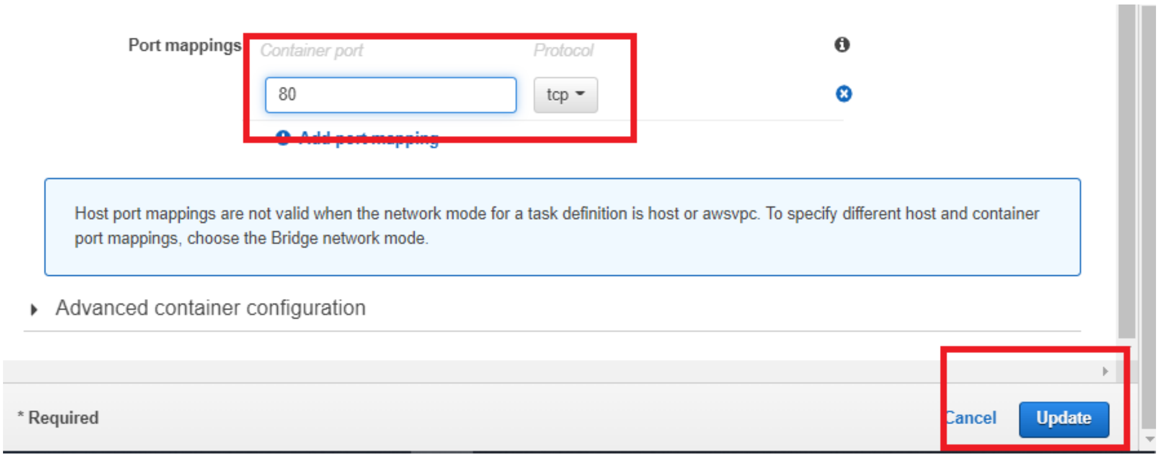
Details

Image tags
latest

URI
450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache:latest

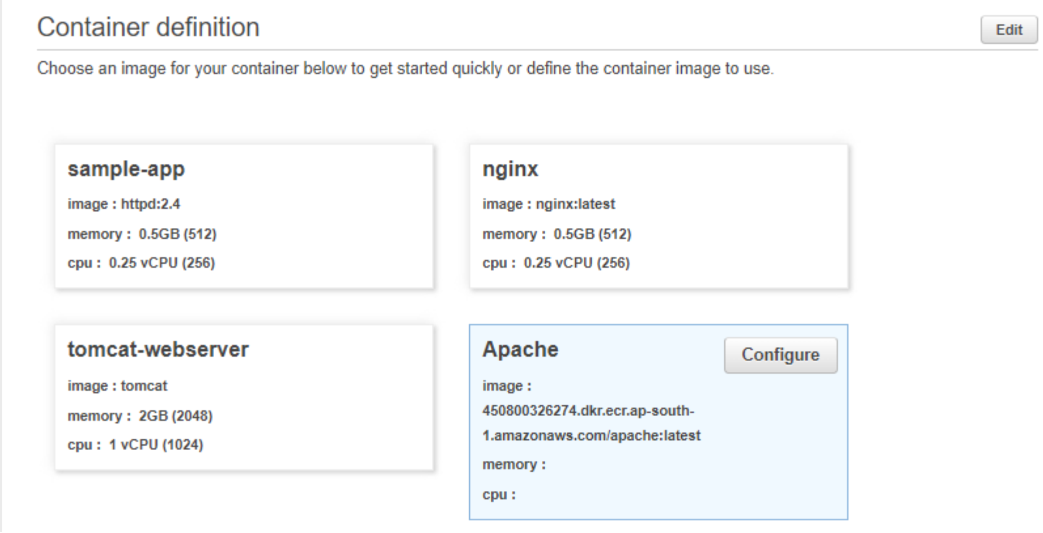
Digest
sha256:fcd3da827eed60495ce3602cb5e04f1c1838af2756660cb8432e1edd6fdbdfdf

4. In port mapping select 80 and click update



Port mappings — Container port: 80 — Protocol: tcp

Host port mappings are not valid when the network mode for a task definition is host or awsvpc. To specify different host and container port mappings, choose the Bridge network mode.



Container definition

Choose an image for your container below to get started quickly or define the container image to use.

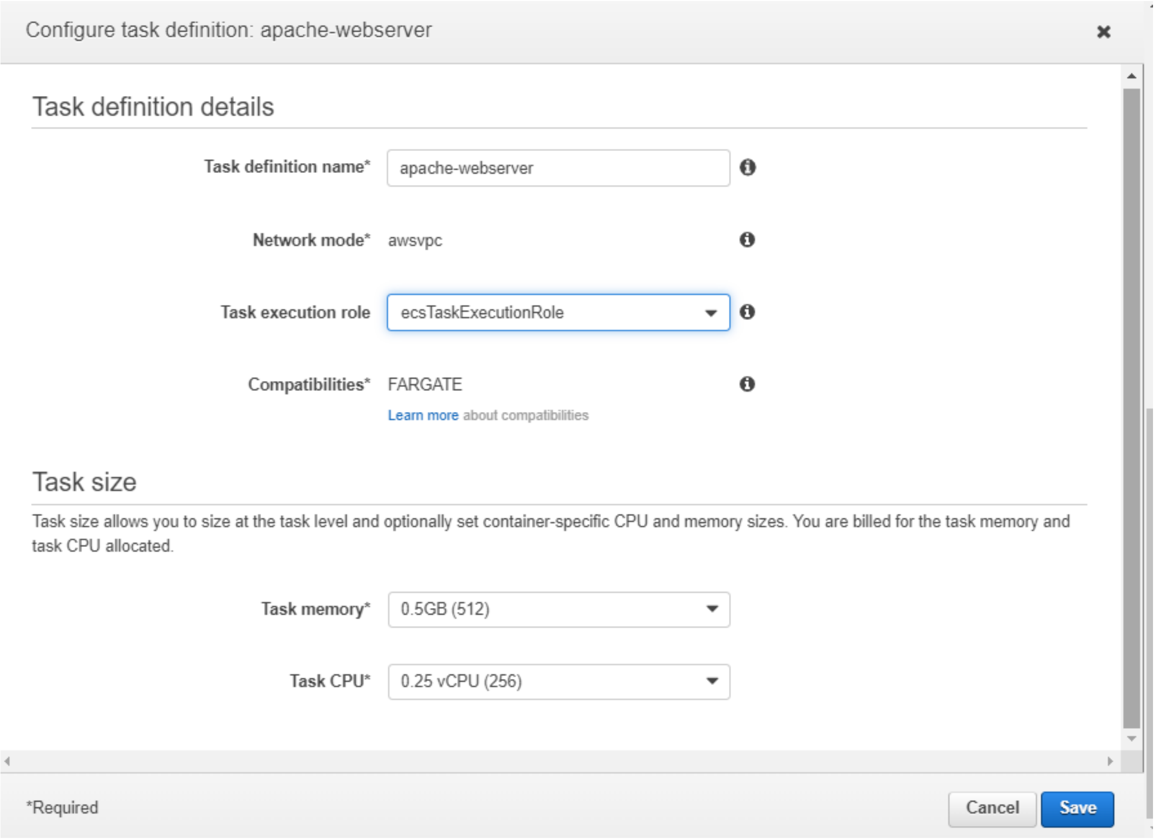
sample-app
image : httpd:2.4
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

nginx
image : nginx:latest
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

tomcat-webserver
image : tomcat
memory : 2GB (2048)
cpu : 1 vCPU (1024)

Apache
image : 450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache:latest
memory :
cpu :

5. In Task defination



Configure task definition: apache-webserver

Task definition details

Task definition name* apache-webserver

Network mode* awsvpc

Task execution role ecsTaskExecutionRole

Compatibilities* FARGATE

Learn more about compatibilities

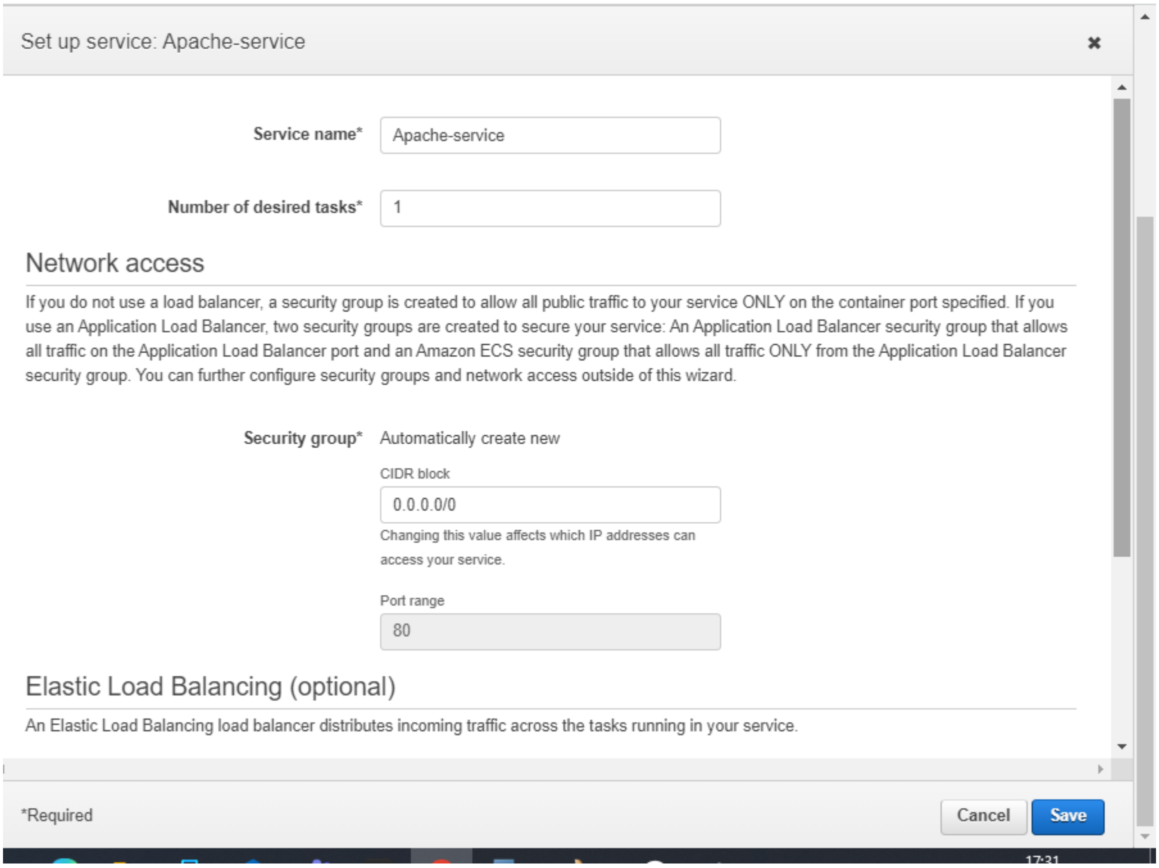
Task size

Task size allows you to size at the task level and optionally set container-specific CPU and memory sizes. You are billed for the task memory and task CPU allocated.

Task memory* 0.5GB (512)

Task CPU* 0.25 vCPU (256)

*Required



Set up service: Apache-service

Service name* Apache-service

Number of desired tasks* 1

Network access

If you do not use a load balancer, a security group is created to allow all public traffic to your service ONLY on the container port specified. If you use an Application Load Balancer, two security groups are created to secure your service: An Application Load Balancer security group that allows all traffic on the Application Load Balancer port and an Amazon ECS security group that allows all traffic ONLY from the Application Load Balancer security group. You can further configure security groups and network access outside of this wizard.

Security group* Automatically create new

CIDR block 0.0.0.0/0

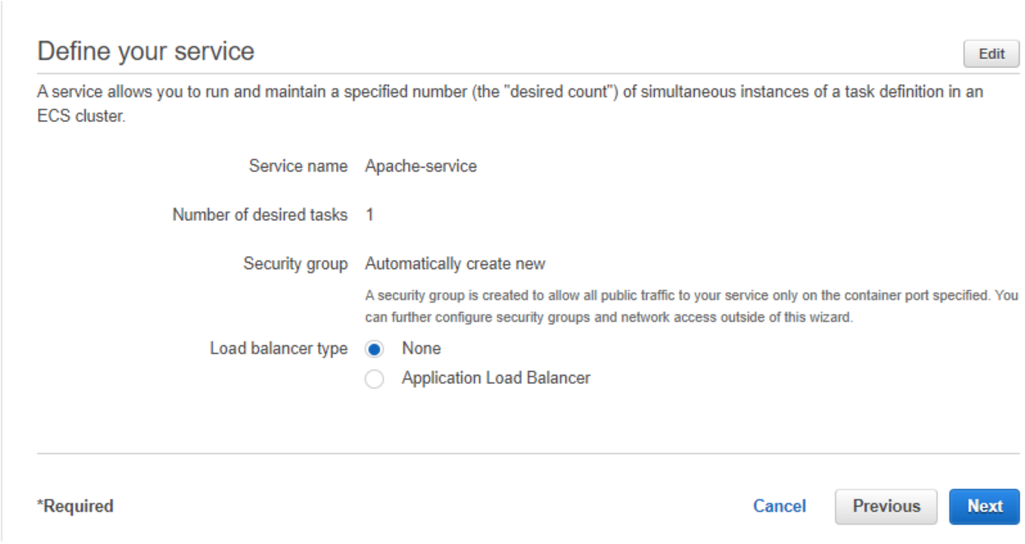
Changing this value affects which IP addresses can access your service.

Port range 80

Elastic Load Balancing (optional)

An Elastic Load Balancing load balancer distributes incoming traffic across the tasks running in your service.

*Required



Define your service

A service allows you to run and maintain a specified number (the "desired count") of simultaneous instances of a task definition in an ECS cluster.

Service name Apache-service

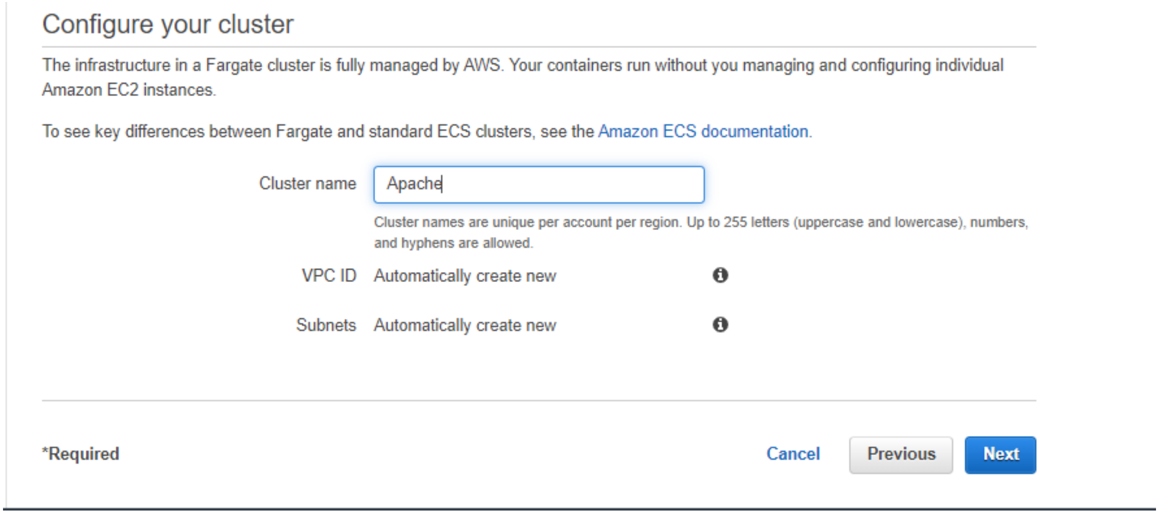
Number of desired tasks 1

Security group Automatically create new

A security group is created to allow all public traffic to your service only on the container port specified. You can further configure security groups and network access outside of this wizard.

Load balancer type ● None
○ Application Load Balancer

*Required



Configure your cluster

The infrastructure in a Fargate cluster is fully managed by AWS. Your containers run without you managing and configuring individual Amazon EC2 instances.

To see key differences between Fargate and standard ECS clusters, see the Amazon ECS documentation.

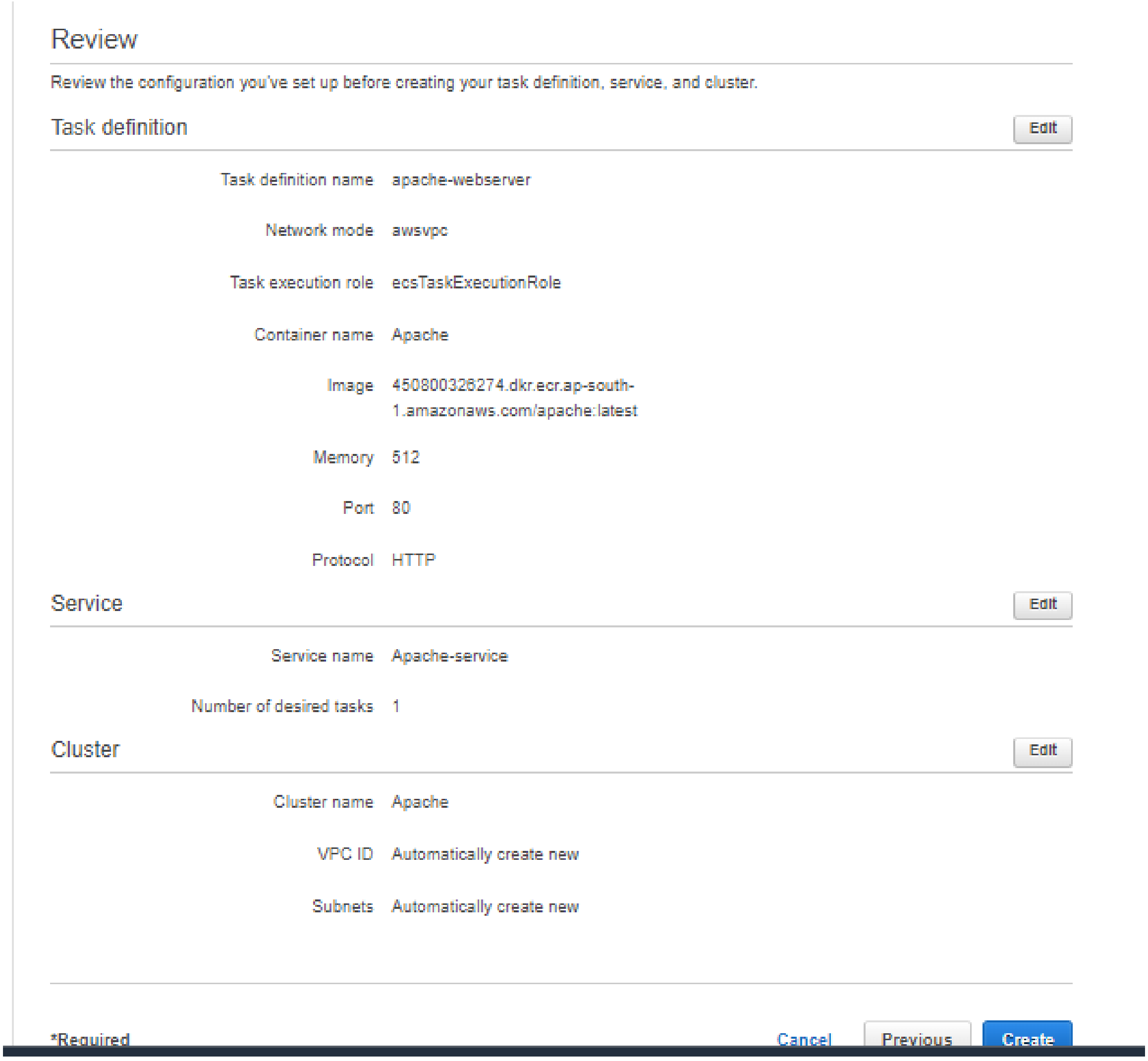
Cluster name Apache

Cluster names are unique per account per region. Up to 255 letters (uppercase and lowercase), numbers, and hyphens are allowed.

VPC ID Automatically create new

Subnets Automatically create new

*Required



Review

Review the configuration you've set up before creating your task definition, service, and cluster.

Task definition

Task definition name	apache-webserver
Network mode	awsvpc
Task execution role	ecsTaskExecutionRole
Container name	Apache
Image	450800326274.dkr.ecr.ap-south-1.amazonaws.com/apache:latest
Memory	512
Port	80
Protocol	HTTP

Service

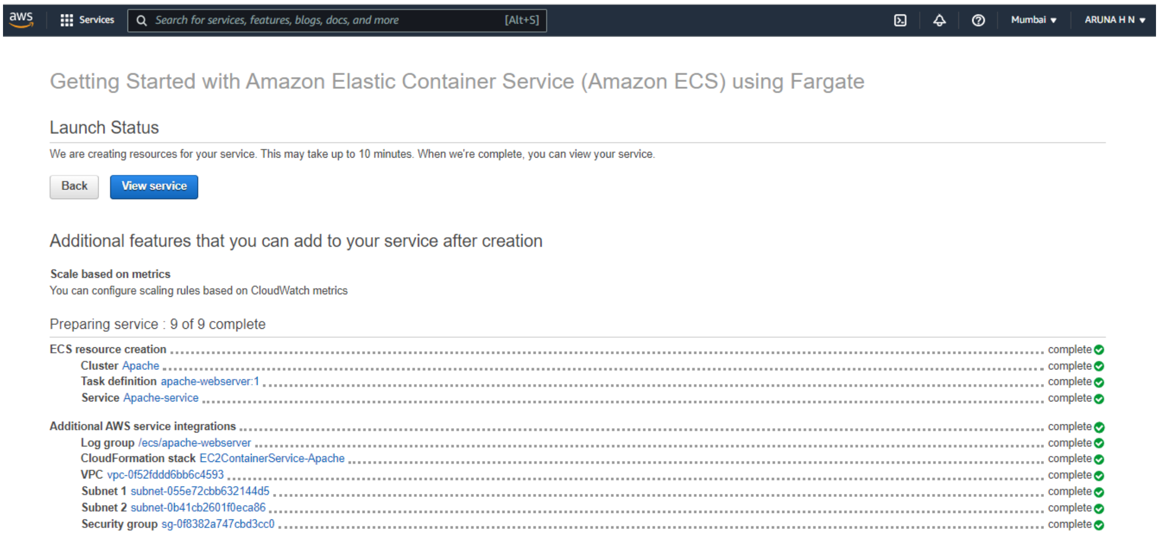
Service name	Apache-service
Number of desired tasks	1

Cluster

Cluster name	Apache
VPC ID	Automatically create new
Subnets	Automatically create new

*Required

5. Wait for Some minutes, It will create the following Resources



aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Mumbai ▾
ARUNA H N ▾

Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

Launch Status

We are creating resources for your service. This may take up to 10 minutes. When we're complete, you can view your service.

Additional features that you can add to your service after creation

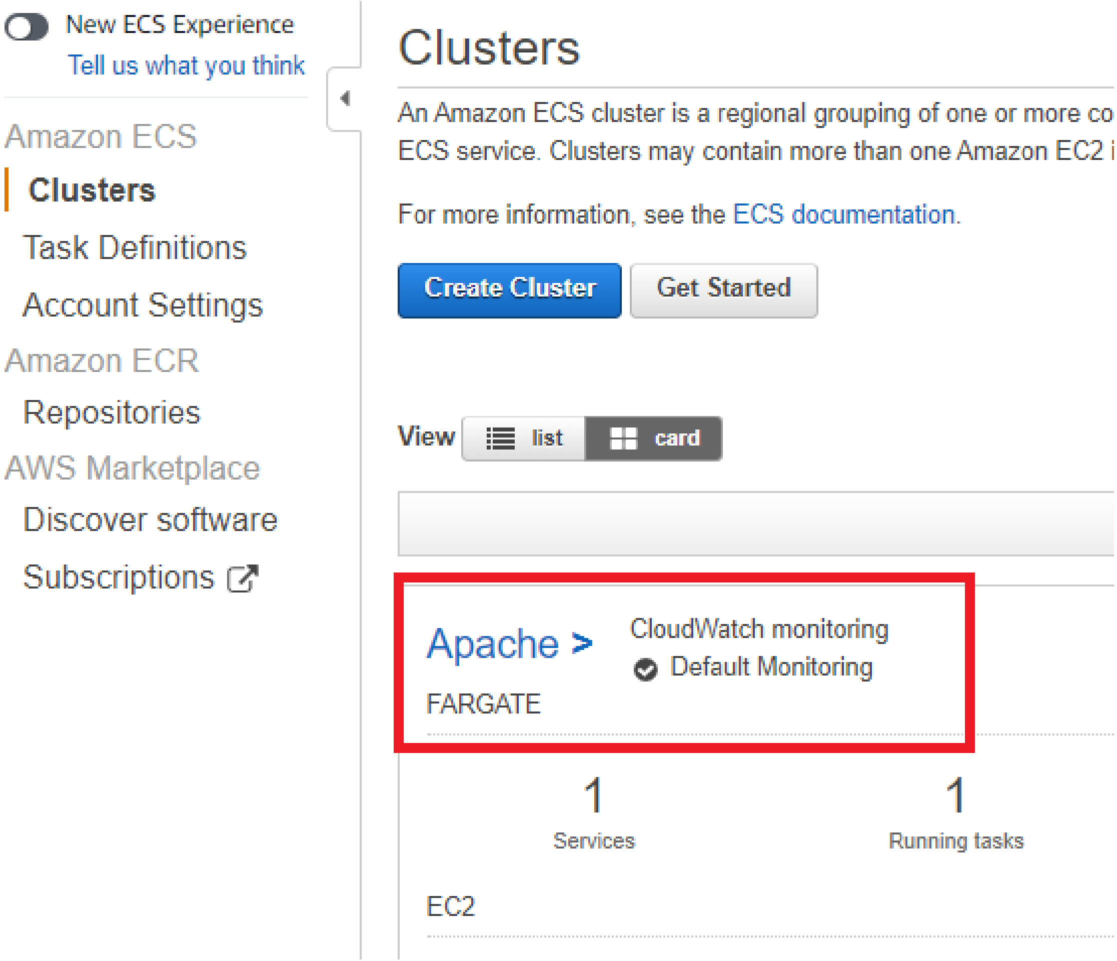
Scale based on metrics
You can configure scaling rules based on CloudWatch metrics

Preparing service : 9 of 9 complete

```
ECS resource creation ........................................ complete ✓
    Cluster Apache ........................................... complete ✓
    Task definition apache-webserver:1 ....................... complete ✓
    Service Apache-service ................................... complete ✓

Additional AWS service integrations .......................... complete ✓
    Log group /ecs/apache-webserver .......................... complete ✓
    CloudFormation stack EC2ContainerService-Apache .......... complete ✓
    VPC vpc-0f52fddd6bb6c4593 ................................ complete ✓
    Subnet 1 subnet-055e72cbb632144d5 ........................ complete ✓
    Subnet 2 subnet-0b41cb2601f0eca86 ........................ complete ✓
    Security group sg-0f8382a747cbd3cc0 ...................... complete ✓
```

6. Goto Clusters select Apache



Clusters

An Amazon ECS cluster is a regional grouping of one or more co
ECS service. Clusters may contain more than one Amazon EC2 i

For more information, see the ECS documentation.

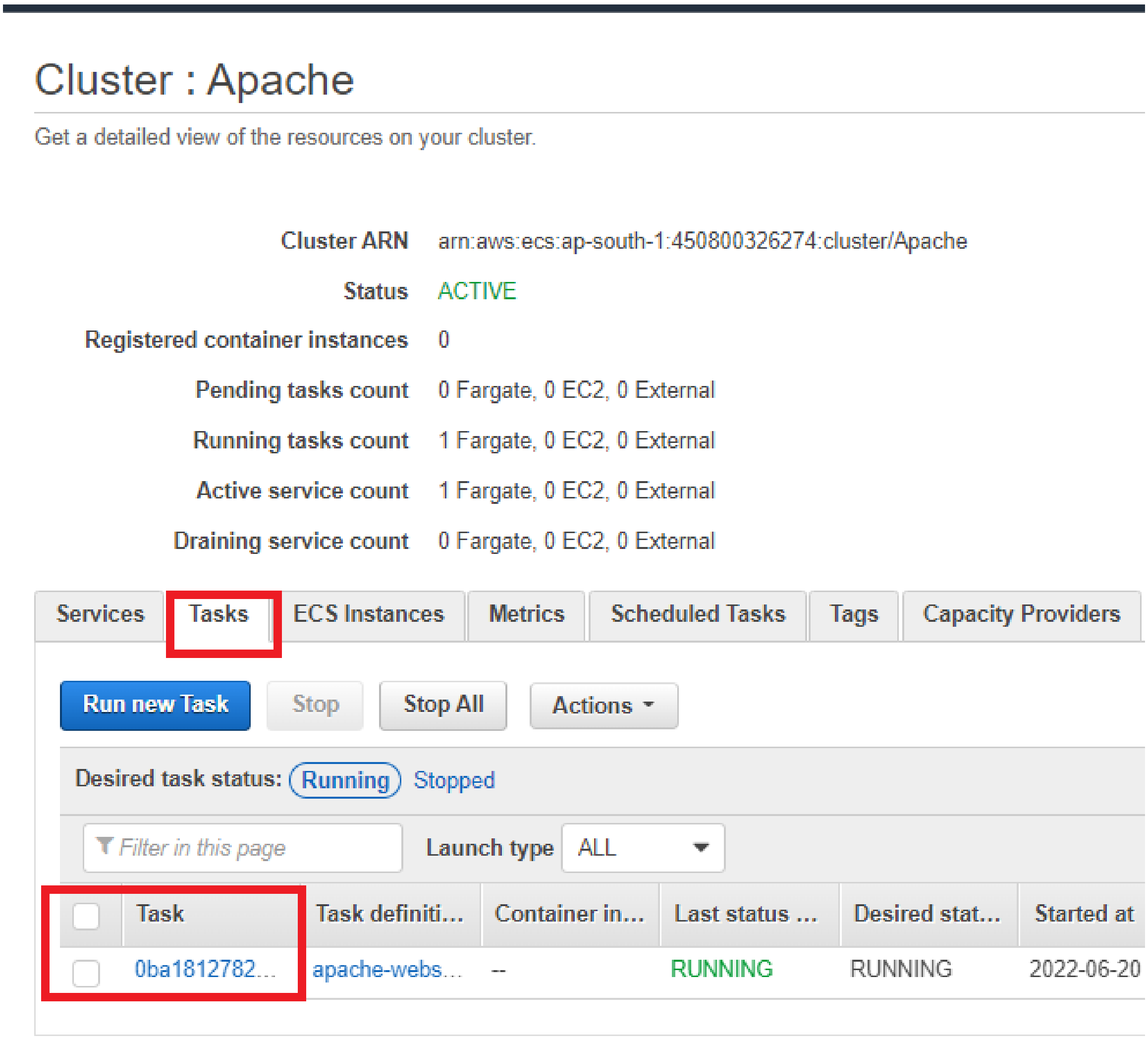
Apache >
FARGATE

CloudWatch monitoring
✓ Default Monitoring

1	1
Services	Running tasks

EC2

Click on Task ID on Task bar



Cluster : Apache

Get a detailed view of the resources on your cluster.

Cluster ARN arn:aws:ecs:ap-south-1:450800326274:cluster/Apache

Status ACTIVE

Registered container instances 0

Pending tasks count 0 Fargate, 0 EC2, 0 External

Running tasks count 1 Fargate, 0 EC2, 0 External

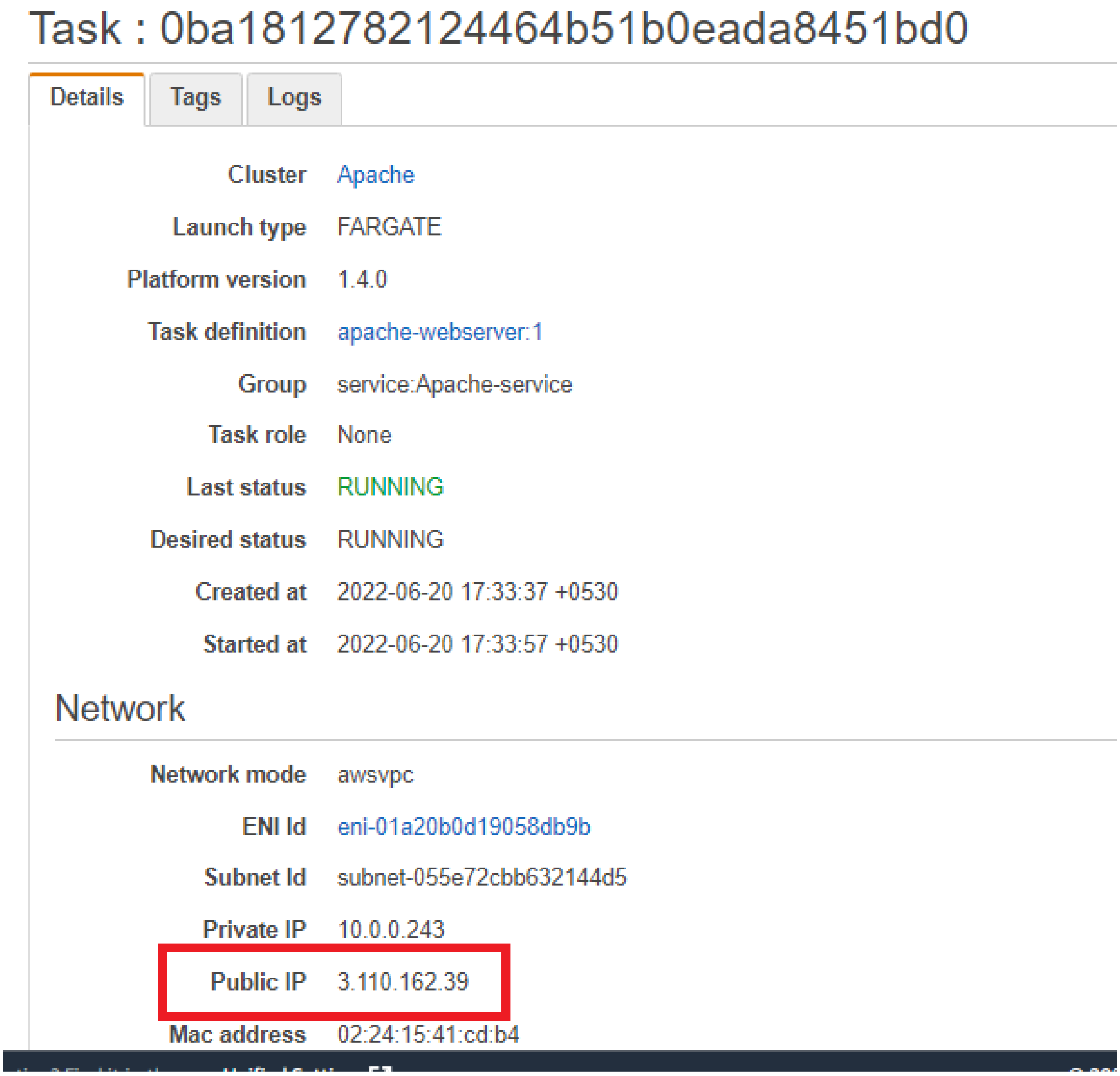
Active service count 1 Fargate, 0 EC2, 0 External

Draining service count 0 Fargate, 0 EC2, 0 External

Desired task status: Running Stopped

Launch type ALL

□	Task	Task definiti...	Container in...	Last status ...	Desired stat...	Started at
□	0ba1812782...	apache-webs...	--	RUNNING	RUNNING	2022-06-20

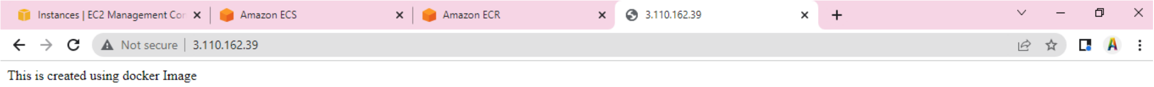


Task : 0ba1812782124464b51b0eada8451bd0

Cluster	Apache
Launch type	FARGATE
Platform version	1.4.0
Task definition	apache-webserver:1
Group	service:Apache-service
Task role	None
Last status	RUNNING
Desired status	RUNNING
Created at	2022-06-20 17:33:37 +0530
Started at	2022-06-20 17:33:57 +0530

Network

Network mode	awsvpc
ENI Id	eni-01a20b0d19058db9b
Subnet Id	subnet-055e72cbb632144d5
Private IP	10.0.0.243
Public IP	3.110.162.39
Mac address	02:24:15:41:cd:b4



This is created using docker Image