CLOUD COMPUTING

Project Report

Student Name	Mail ID	Social Security Number
Namburu Sri Harsha	srna21@student.bth.se	20001227-T235
Suvoj Reddy Kovvuri	Suko21@student.bth.se	20010306-T032
Lydia Sri Divya Dommeti	Lydo21@student.bth.se	20000915-T366

Introduction:

This is a flask application, a python based one. It is usually a simple and basic face detection web application in which user can upload an image and we will get an output detecting the faces.

Out main motive is to implement the application scalable and fault tolerant.

We also wanted to deploy in a cluster, because, we will get more power rather than that of a single standalone machine.

Steps:

- Create the application
- Dockerize it
- Push the image to AWS ECR
- Create a ECS cluster
- Define a task definition
- Open the cluster and create a service
- While creating the service and task, create and configure auto-scaling and load balancing as well
- Wait for the status changing to RUNNING state
- Copy the IP address and run on the machine

Implementation:

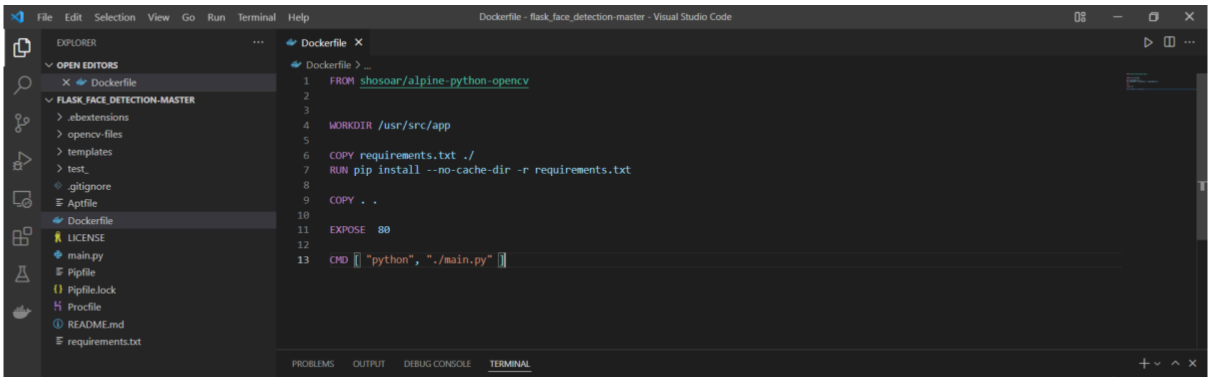
Step 1:

The application was made firstly

We then have developed a Dockerfile in which the commands to run will be there.

By building this file, we will get a customized image that contains our web facial detection flask application.

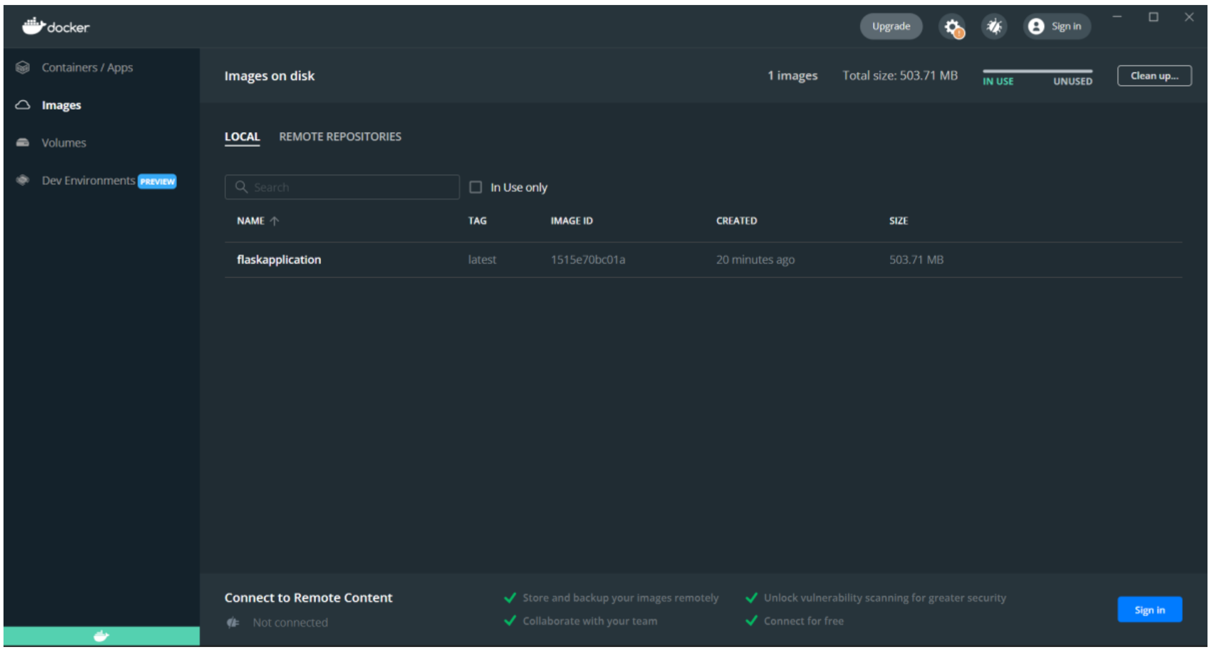
The following is the docker file that we have wrote for our flask application.



```
FROM shosoar/alpine-python-opencv


WORKDIR /usr/src/app

COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

EXPOSE  80

CMD [ "python", "./main.py" ]
```

Step 2:

After building the application, we will get to see the image that have been build in the docker application as well



Images on disk

1 images Total size: 503.71 MB IN USE UNUSED

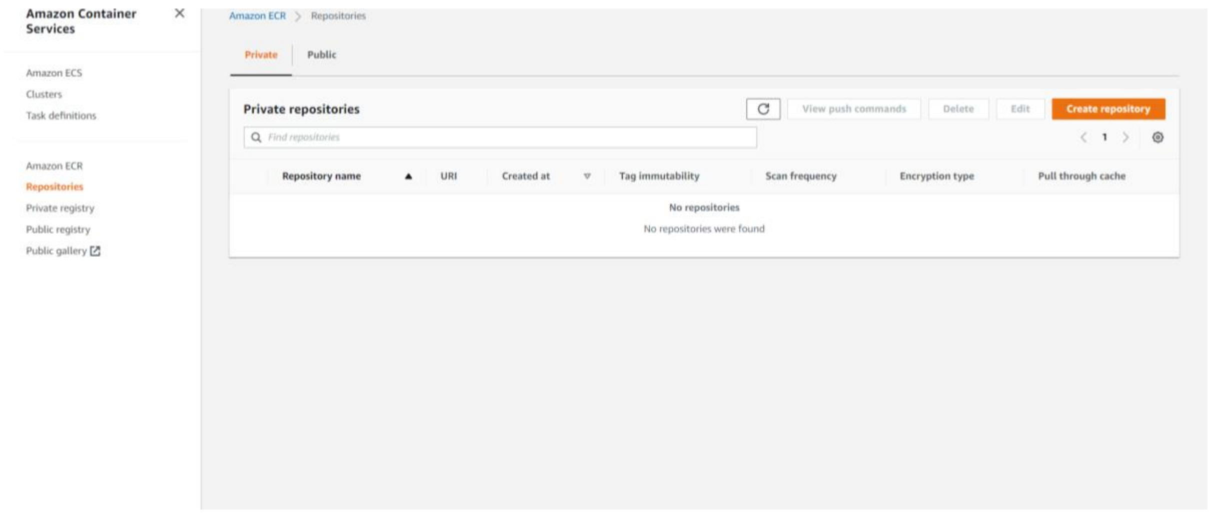
NAME	TAG	IMAGE ID	CREATED	SIZE
flaskapplication	latest	1515e70bc01a	20 minutes ago	503.71 MB

Connect to Remote Content
Not connected
✓ Store and backup your images remotely
✓ Collaborate with your team
✓ Unlock vulnerability scanning for greater security
✓ Connect for free

Step 3:

In the AWS account, a repository should be created in the ECR service offered by the Aws itself

The AWS provided all the things in GUI manner, so, it became easy to deploy the application;

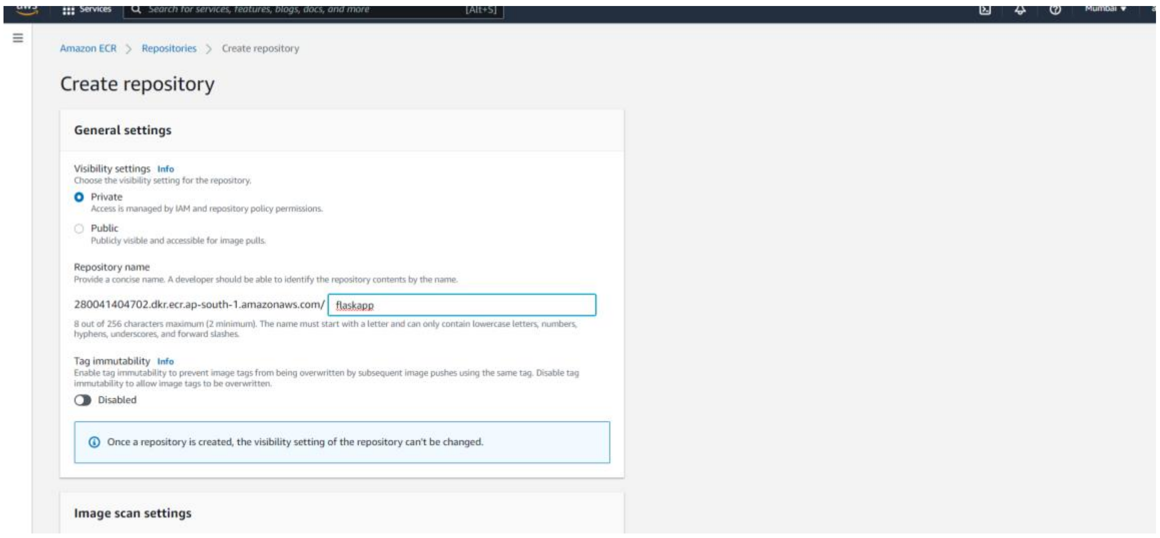


Private repositories

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

No repositories
No repositories were found

We have created our repository in AWS as discussed above


Services
Search for services, features, blogs, docs, and more [Alt+S]
Mumbai ▾
Amazon ECR > Repositories > Create repository

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.

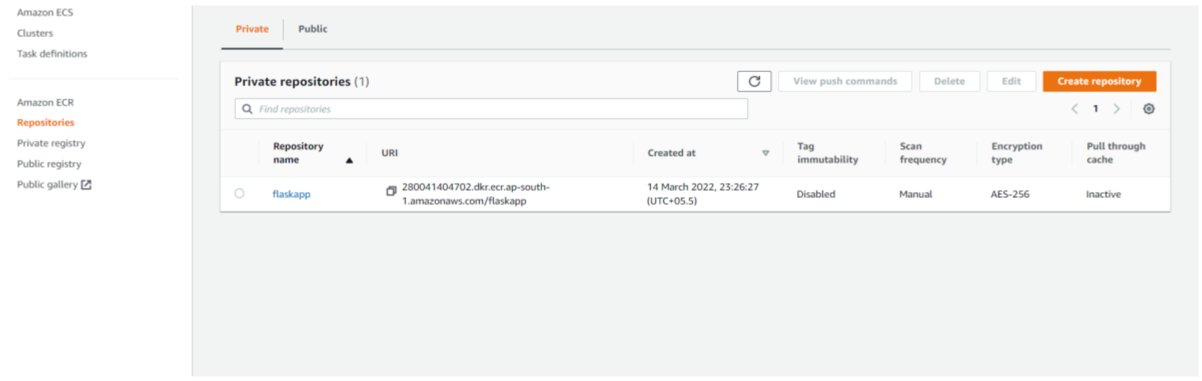
Repository name
Provide a concise name. A developer should be able to identify the repository contents by the name.
280041404702.dkr.ecr.ap-south-1.amazonaws.com/ flaskapp
8 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, 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

The repository is ready.

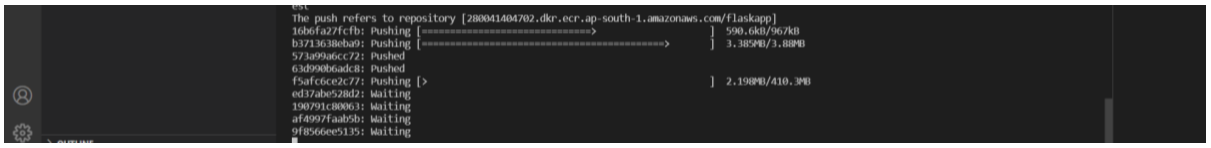


Private repositories (1)

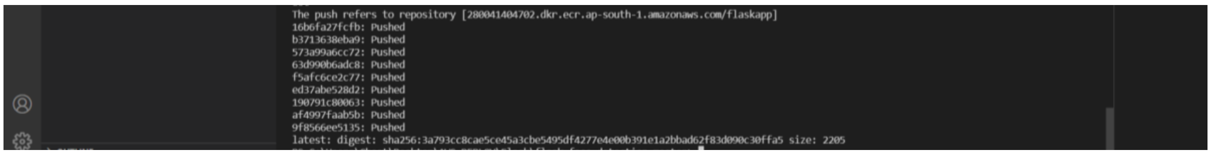
	Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type	Pull through cache
○	flaskapp	280041404702.dkr.ecr.ap-south-1.amazonaws.com/flaskapp	14 March 2022, 23:26:27 (UTC+05.5)	Disabled	Manual	AES-256	Inactive

Step 4:

Now the application should be pushed to the Aws repository.



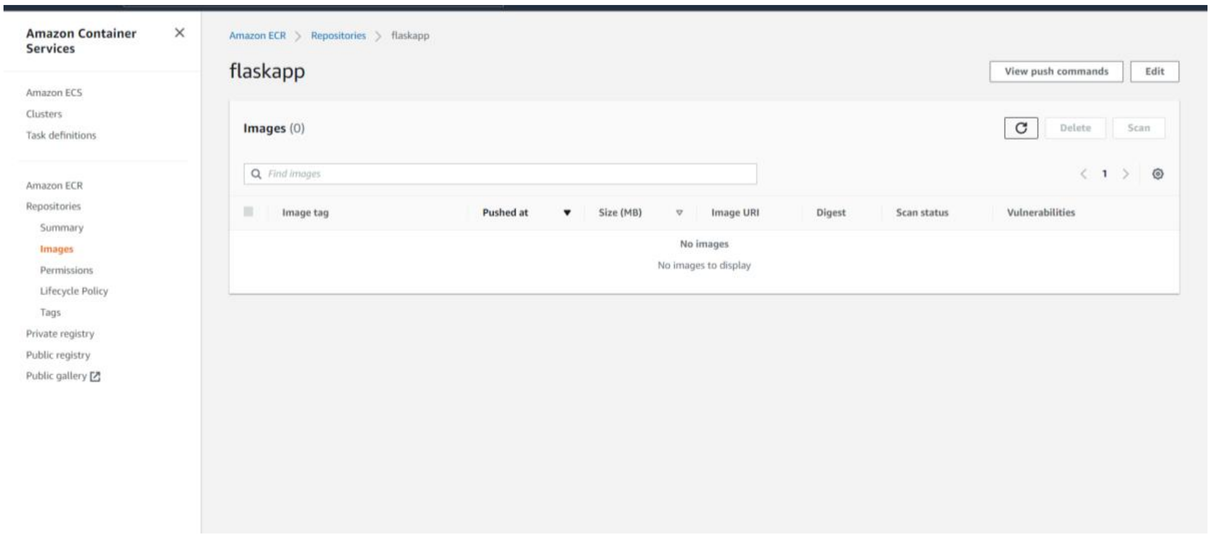
```
The push refers to repository [280041404702.dkr.ecr.ap-south-1.amazonaws.com/flaskapp]
16b6fa27fcfb: Pushing [==============================>                    ]  590.6kB/967kB
b3713638eba9: Pushing [============================================>      ]  3.385MB/3.88MB
573a99a6cc72: Pushed
63d990b6adc8: Pushed
f5afc6ce2c77: Pushing [>                                                  ]  2.198MB/410.3MB
ed37abe528d2: Waiting
190791c80063: Waiting
af4997faab5b: Waiting
9f8566ee5135: Waiting
```



```
The push refers to repository [280041404702.dkr.ecr.ap-south-1.amazonaws.com/flaskapp]
16b6fa27fcfb: Pushed
b3713638eba9: Pushed
573a99a6cc72: Pushed
63d990b6adc8: Pushed
f5afc6ce2c77: Pushed
ed37abe528d2: Pushed
190791c80063: Pushed
af4997faab5b: Pushed
9f8566ee5135: Pushed
latest: digest: sha256:3a793cc8cae5ce45a3cbe5495df4277e4e00b391e1a2bbad62f83d090c30ffa5 size: 2205
```

After successful push, we will get to see this

The below screenshot is the empty repository (before pushing)


Amazon Container Services

Amazon ECS
Clusters
Task definitions

Amazon ECR
Repositories
Summary
Images
Permissions
Lifecycle Policy
Tags
Private registry
Public registry
Public gallery

Amazon ECR > Repositories > flaskapp

flaskapp

View push commands | Edit

Images (0)

Delete | Scan

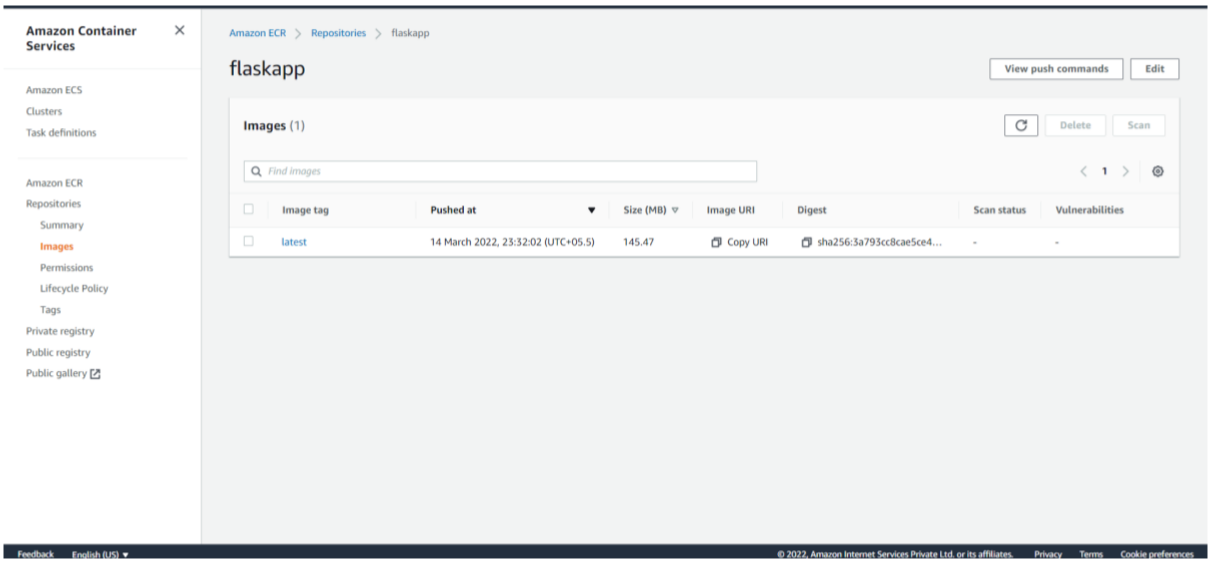
Find images

Image tag	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulnerabilities

No images
No images to display

We can see our image that we have pushed from docker in the AWS repo.

This is the latest push we have done so far.


Amazon Container Services

Amazon ECS
Clusters
Task definitions

Amazon ECR
Repositories
Summary
Images
Permissions
Lifecycle Policy
Tags
Private registry
Public registry
Public gallery

Amazon ECR > Repositories > flaskapp

flaskapp

View push commands | Edit

Images (1)

Delete | Scan

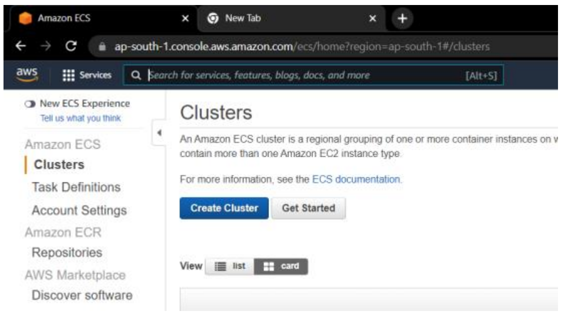
Find images

Image tag	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulnerabilities
latest	14 March 2022, 23:32:02 (UTC+05.5)	145.47	Copy URI	sha256:3a793cc8cae5ce4...	-	-

Feedback English (US) ▾ © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Step 5:

We had created an cluster using the ECS service that is beeing offered by AWS


Amazon ECS | New Tab

ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters

aws Services Search for services, features, blogs, docs, and more [Alt+S]

New ECS Experience
Tell us what you think

Amazon ECS
Clusters
Task Definitions
Account Settings
Amazon ECR
Repositories
AWS Marketplace
Discover software

Clusters

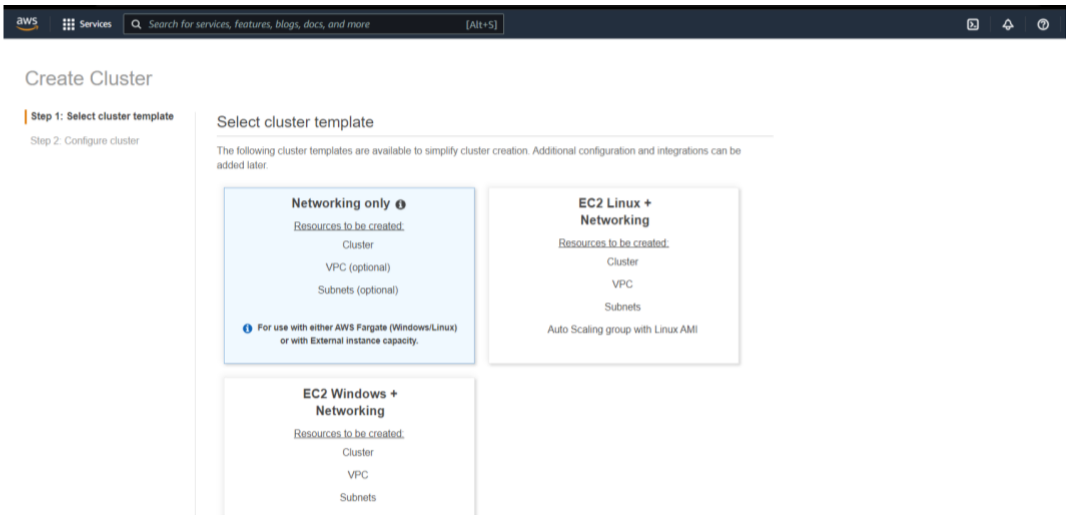
An Amazon ECS cluster is a regional grouping of one or more container instances on v
contain more than one Amazon EC2 instance type.

For more information, see the ECS documentation.

Create Cluster | Get Started

View list card

We opted for Networking only because, we will get more options to explore


Create Cluster

Step 1: Select cluster template
Step 2: Configure cluster

Select cluster template

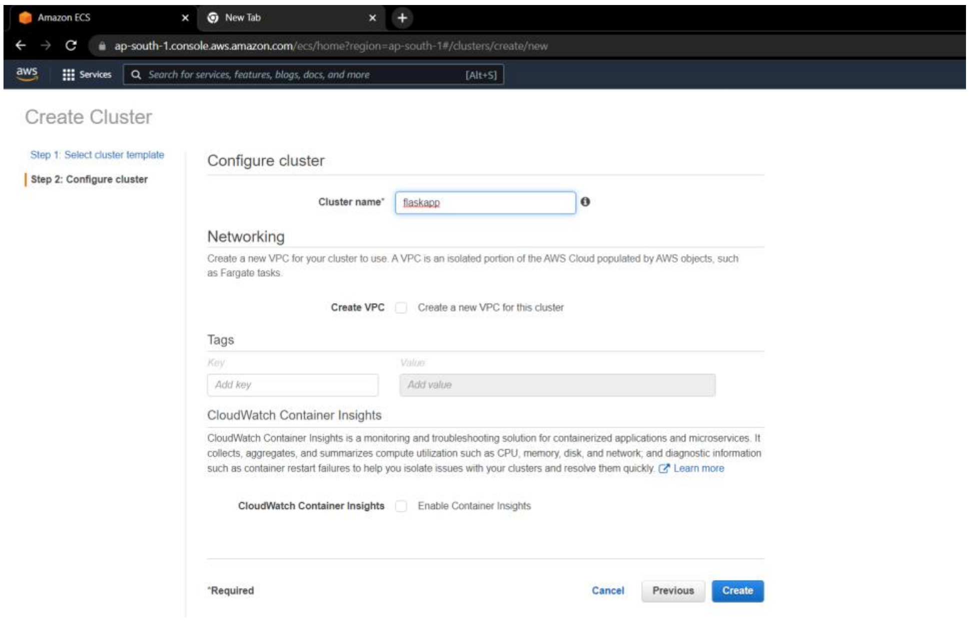
The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.

Networking only
Resources to be created:
Cluster
VPC (optional)
Subnets (optional)
For use with either AWS Fargate (Windows/Linux) or with External instance capacity.

EC2 Linux + Networking
Resources to be created:
Cluster
VPC
Subnets
Auto Scaling group with Linux AMI

EC2 Windows + Networking
Resources to be created:
Cluster
VPC
Subnets

After the previous step, we just have added few configurations to the cluster


ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters/create/new

Create Cluster

Step 1: Select cluster template
Step 2: Configure cluster

Configure cluster

Cluster name* flaskapp

Networking

Create a new VPC for your cluster to use. A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Fargate tasks.

Create VPC ☐ Create a new VPC for this cluster

Tags

Key: Add key Value: Add value

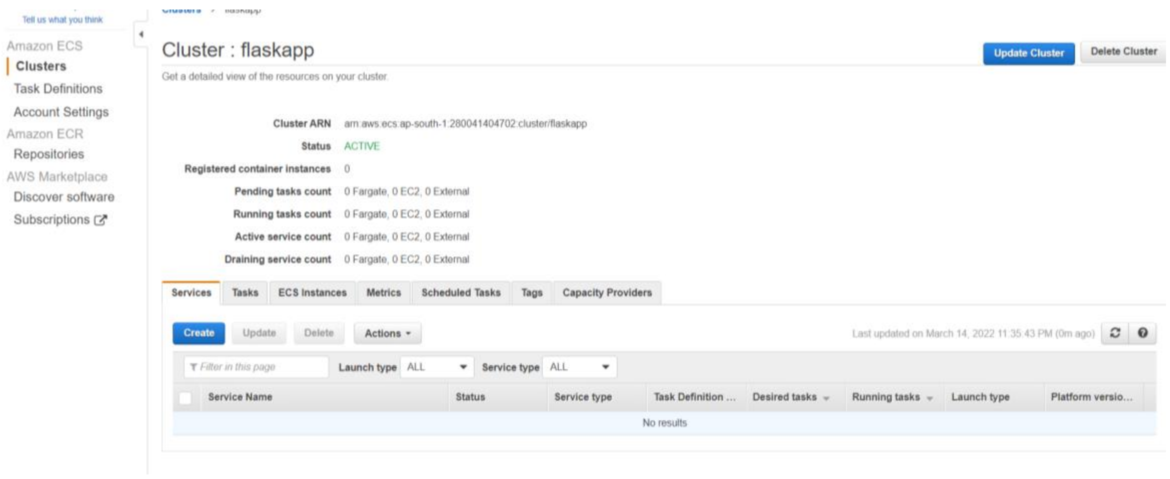
CloudWatch Container Insights

CloudWatch Container Insights is a monitoring and troubleshooting solution for containerized applications and microservices. It collects, aggregates, and summarizes compute utilization such as CPU, memory, disk, and network; and diagnostic information such as container restart failures to help you isolate issues with your clusters and resolve them quickly. Learn more

CloudWatch Container Insights ☐ Enable Container Insights

*Required

At the end, we got a new cluster up and running on AWS


Cluster : flaskapp

Get a detailed view of the resources on your cluster.

Cluster ARN	arn:aws:ecs:ap-south-1:280041404702:cluster/flaskapp
Status	ACTIVE
Registered container instances	0
Pending tasks count	0 Fargate, 0 EC2, 0 External
Running tasks count	0 Fargate, 0 EC2, 0 External
Active service count	0 Fargate, 0 EC2, 0 External
Draining service count	0 Fargate, 0 EC2, 0 External

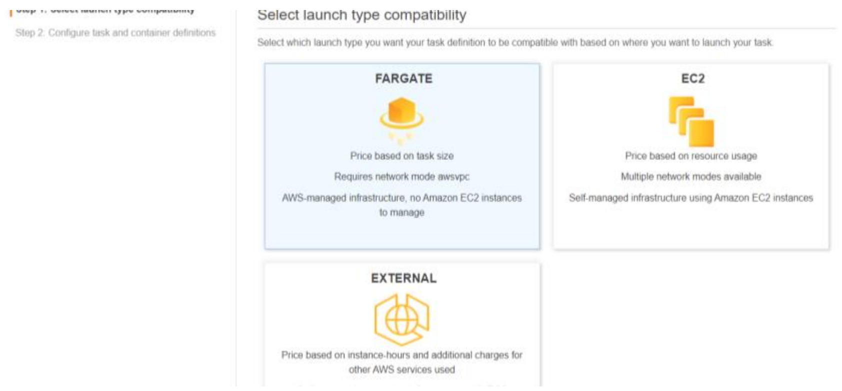
Last updated on March 14, 2022 11:35:43 PM (0m ago)

Launch type: ALL Service type: ALL

Service Name	Status	Service type	Task Definition ...	Desired tasks	Running tasks	Launch type	Platform versio...
No results							

We had also to create a task

We opted for an auto management service that can be attached to ECS deployment, i.e., Fargate

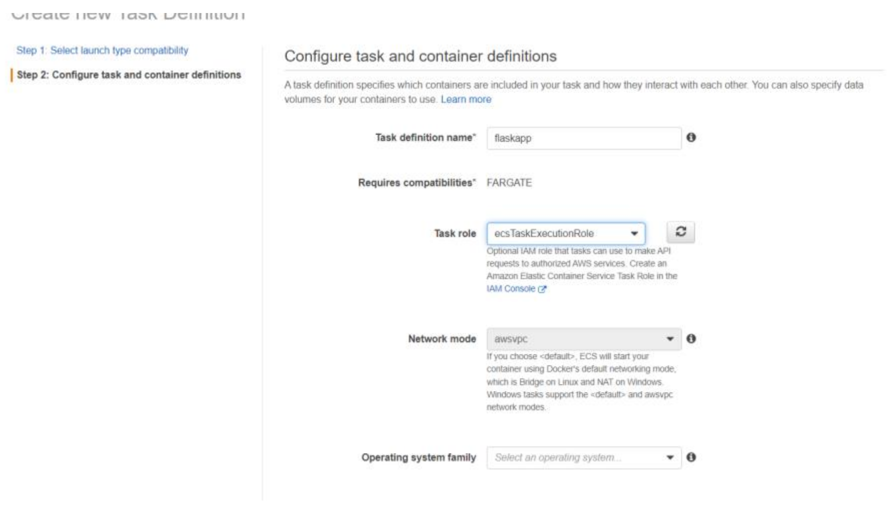

Step 2: Configure task and container definitions

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

FARGATE
Price based on task size
Requires network mode awsvpc
AWS-managed infrastructure, no Amazon EC2 instances to manage
EC2
Price based on resource usage
Multiple network modes available
Self-managed infrastructure using Amazon EC2 instances
EXTERNAL
Price based on instance-hours and additional charges for other AWS services used

SO, now we also started configuring the task definition


Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Configure task and container definitions

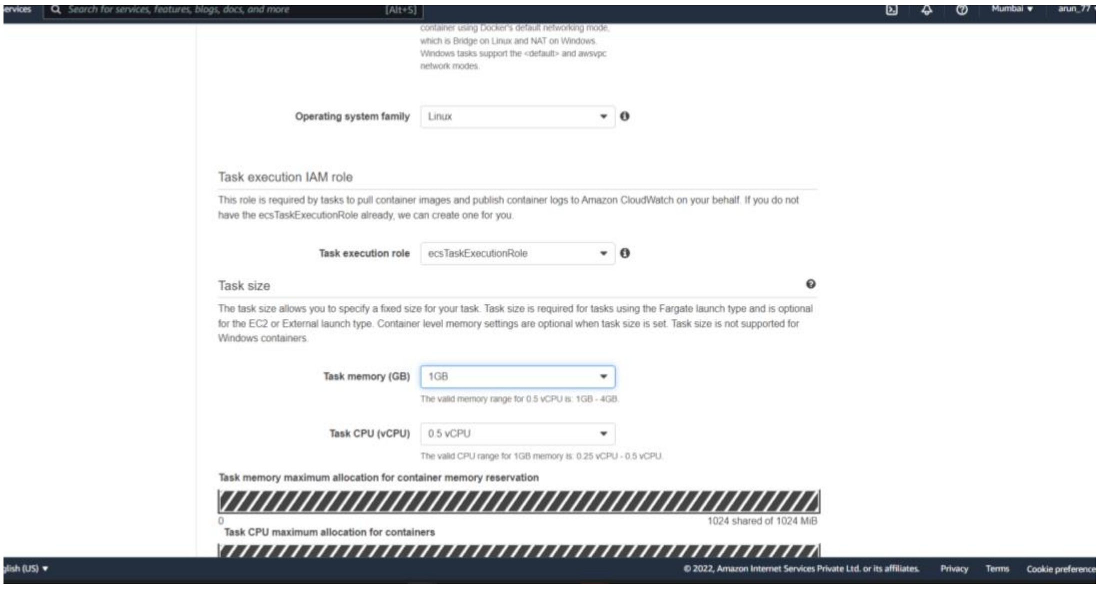
A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. Learn more

Task definition name*	flaskapp
Requires compatibilities*	FARGATE
Task role	ecsTaskExecutionRole
Network mode	awsvpc
Operating system family	Select an operating system...

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the IAM Console

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. Windows tasks support the <default> and awsvpc network modes.

All the important configurations needed to be done


container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. Windows tasks support the <default> and awsvpc network modes.

Operating system family	Linux

Task execution IAM role

This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the ecsTaskExecutionRole already, we can create one for you.

Task execution role	ecsTaskExecutionRole

Task size

The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 or External launch type. Container level memory settings are optional when task size is set. Task size is not supported for Windows containers.

Task memory (GB)	1GB
Task CPU (vCPU)	0.5 vCPU

The valid memory range for 0.5 vCPU is: 1GB - 4GB.

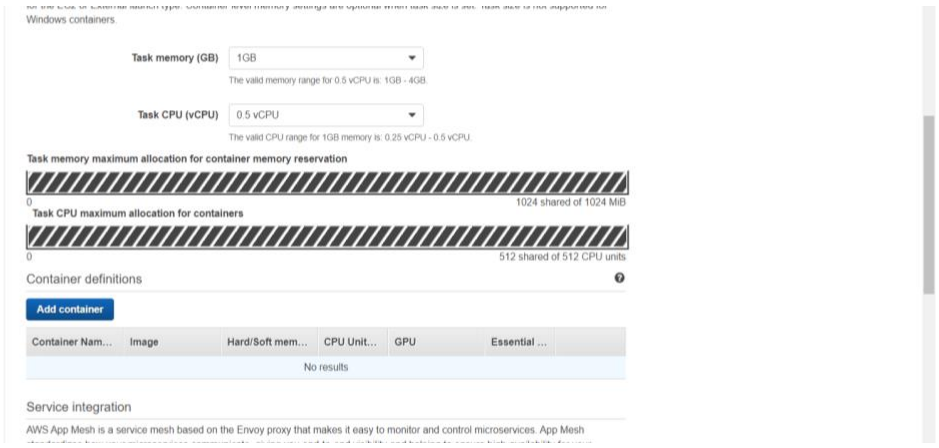
The valid CPU range for 1GB memory is: 0.25 vCPU - 0.5 vCPU.

Task memory maximum allocation for container memory reservation

0 — 1024 shared of 1024 MiB

Task CPU maximum allocation for containers

Search for services, features, blogs, docs, and more
[Alt+S]
Mumbai
arun_77
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preference



Windows containers.

Task memory (GB): 1GB

The valid memory range for 0.5 vCPU is: 1GB - 4GB.

Task CPU (vCPU): 0.5 vCPU

The valid CPU range for 1GB memory is: 0.25 vCPU - 0.5 vCPU.

Task memory maximum allocation for container memory reservation

0 — 1024 shared of 1024 MiB

Task CPU maximum allocation for containers

0 — 512 shared of 512 CPU units

Container definitions

Add container

Container Nam...	Image	Hard/Soft mem...	CPU Unit...	GPU	Essential ...
No results					

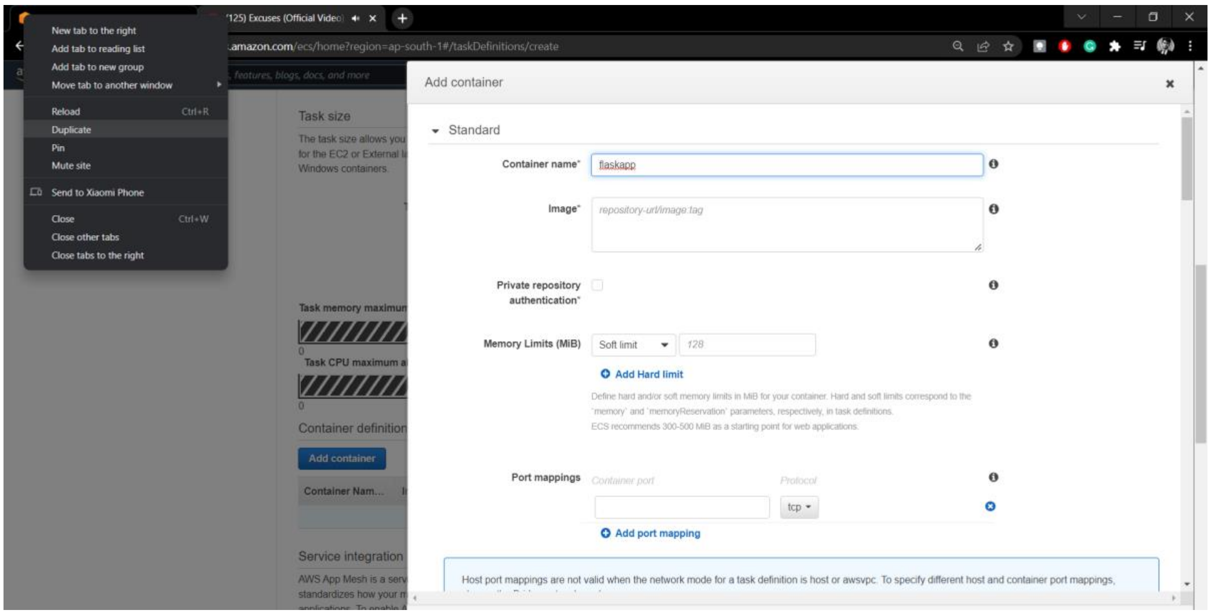
Service integration

AWS App Mesh is a service mesh based on the Envoy proxy that makes it easy to monitor and control microservices. App Mesh

We need to add the container to the task definition

We should select the container that we have uploaded in the AWS container repository

We just need to add the container link to this window



.amazon.com/ecs/home?region=ap-south-1#/taskDefinitions/create

Add container

Standard

Container name*: flaskapp

Image*: repository-url/image:tag

Private repository authentication*

Memory Limits (MiB): Soft limit 128

Add Hard limit

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.
ECS recommends 300-500 MiB as a starting point for web applications.

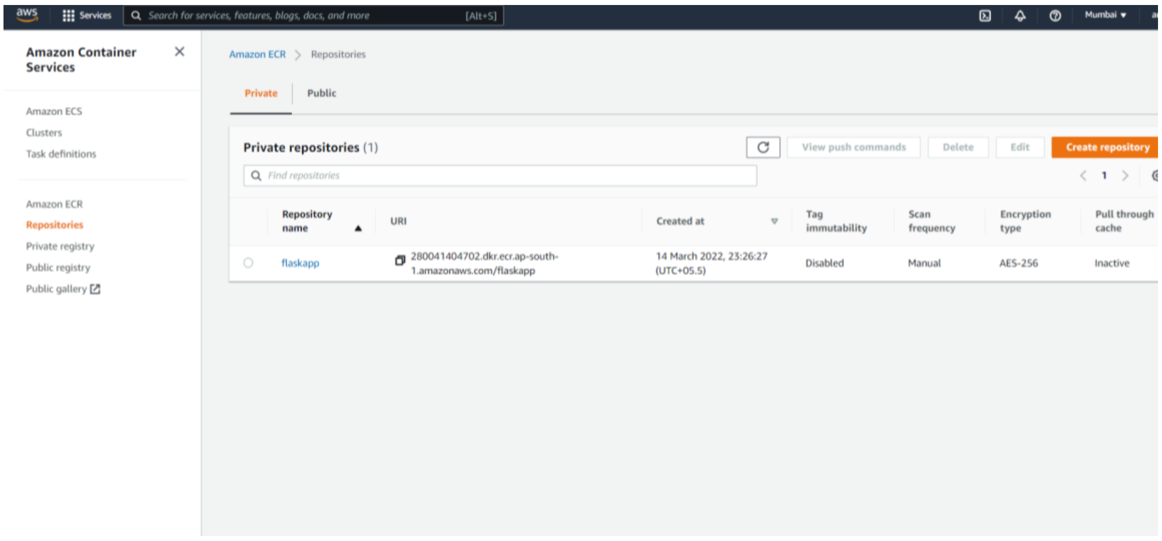
Port mappings: Container port — Protocol: tcp

Add port mapping

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,

We can get the link from the repo.

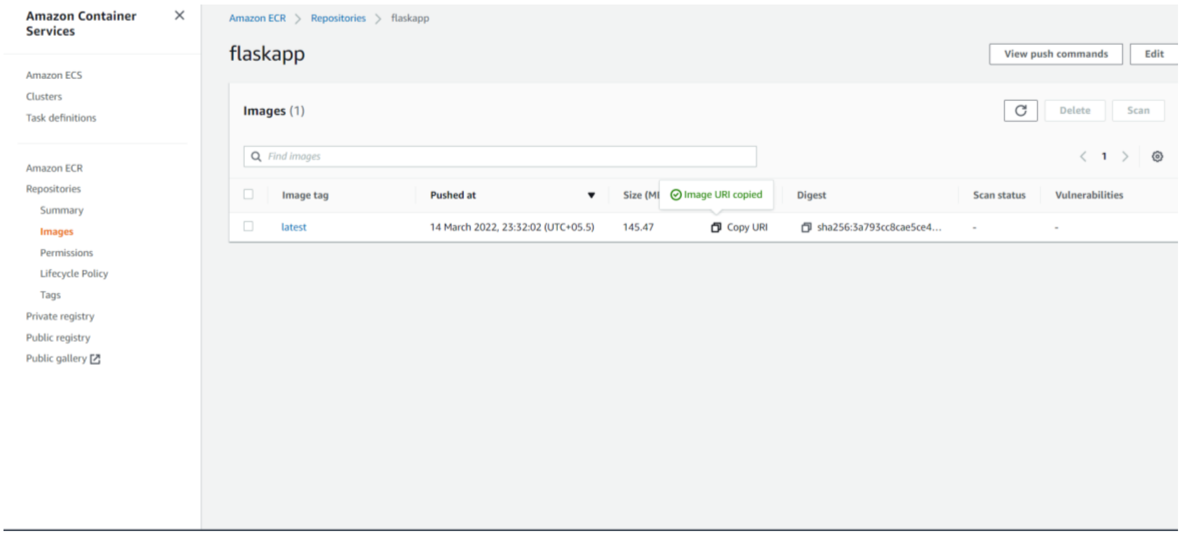
Open the repo just by clicking on it.



Private repositories (1)

Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type	Pull through cache
flaskapp	280041404702.dkr.ecr.ap-south-1.amazonaws.com/flaskapp	14 March 2022, 23:26:27 (UTC+05.5)	Disabled	Manual	AES-256	Inactive

Here we have the copy URI option. We just need to click on it



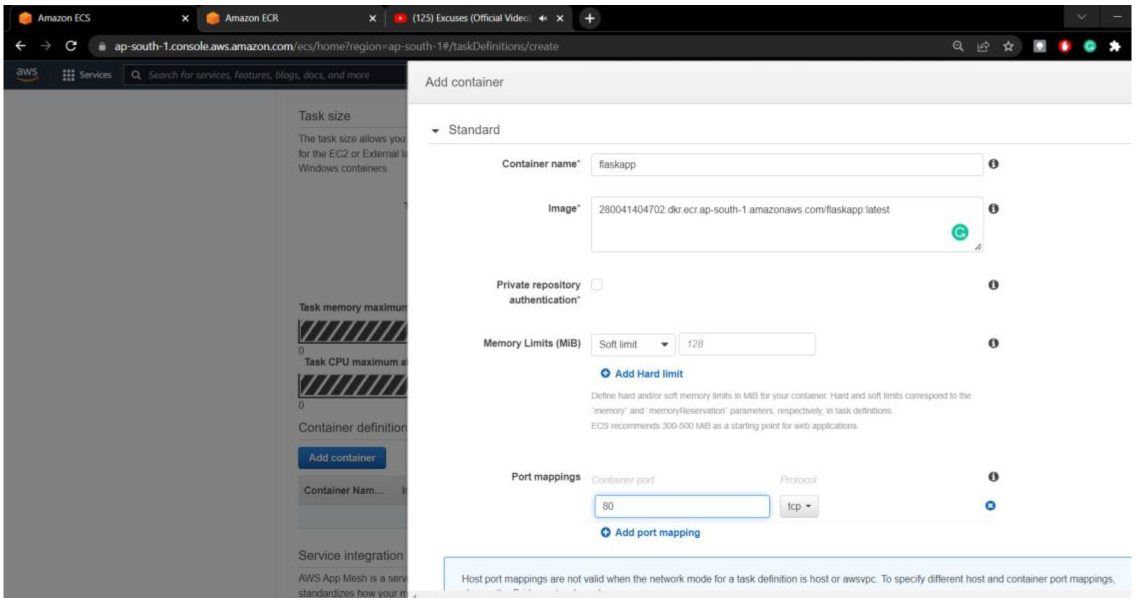
flaskapp

Images (1)

Image tag	Pushed at	Size (MB)	URI	Digest	Scan status	Vulnerabilities
latest	14 March 2022, 23:32:02 (UTC+05.5)	145.47	Copy URI	sha256:3a793cc8cae5ce4...	-	-

Paste the link in the image textbox

Do the port mappings as required


ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/taskDefinitions/create

Add container

Standard

Container name* flaskapp

Image* 280041404702.dkr.ecr.ap-south-1.amazonaws.com/flaskapp:latest

Private repository authentication*

Memory Limits (MiB) Soft limit 128

Add Hard limit

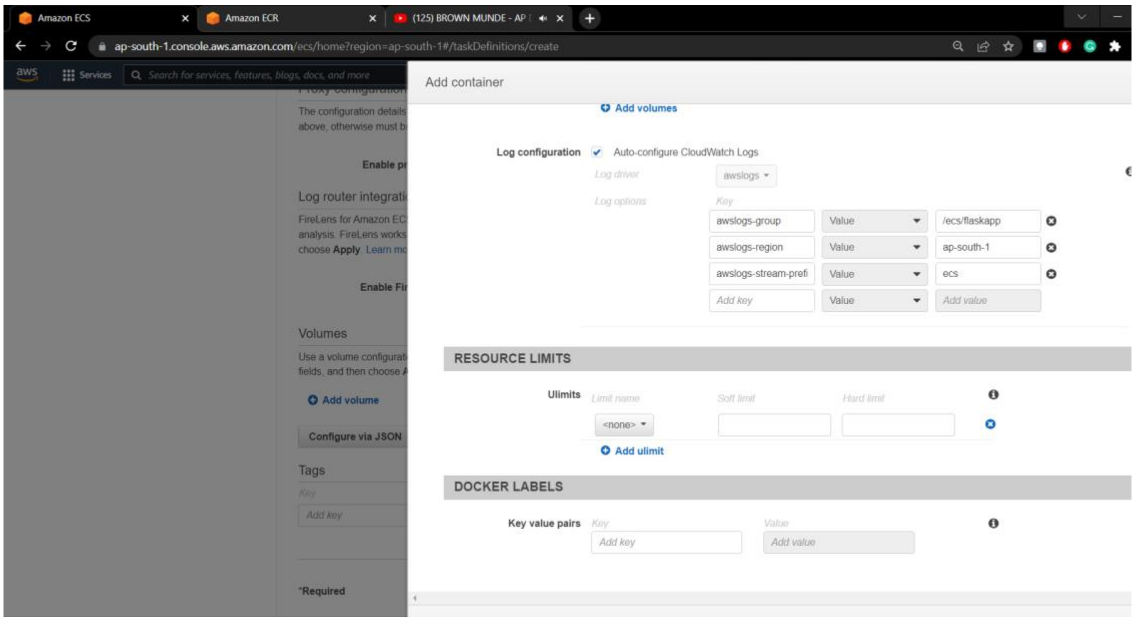
Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, in task definitions.
ECS recommends 300-500 MiB as a starting point for web applications.

Port mappings Container port 80 Protocol tcp

Add port mapping

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,

Do the other configurations if they are required


ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/taskDefinitions/create

Add container

Add volumes

Log configuration ✓ Auto-configure CloudWatch Logs

Log driver awslogs

Log options

Key		
awslogs-group	Value	/ecs/flaskapp
awslogs-region	Value	ap-south-1
awslogs-stream-prefi	Value	ecs
Add key	Value	Add value

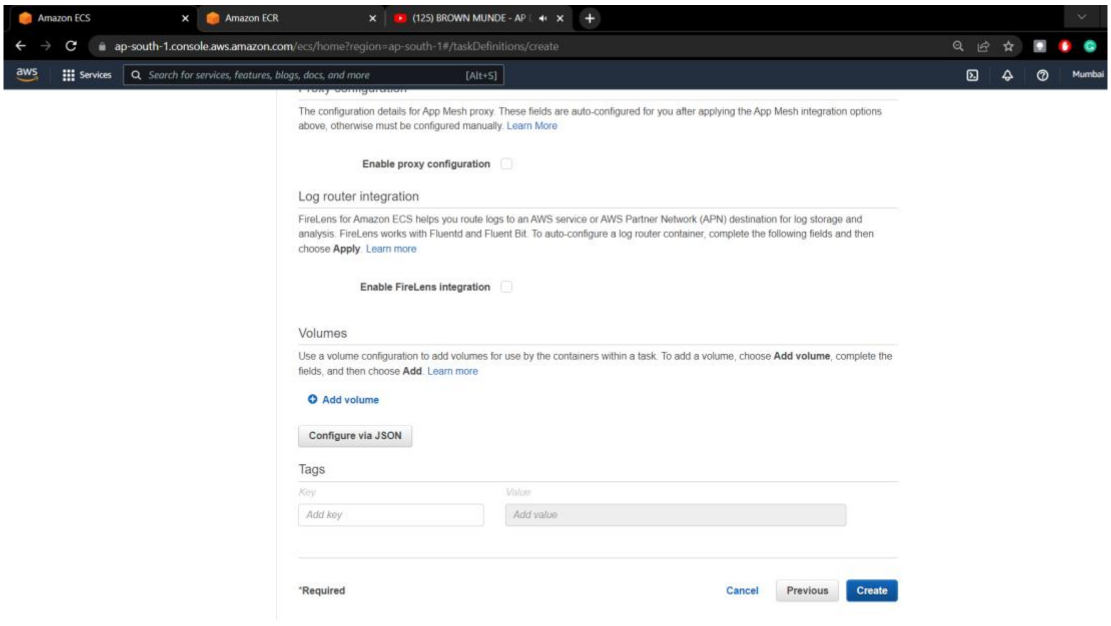
RESOURCE LIMITS

Ulimits Limit name <none> Soft limit Hard limit

Add ulimit

DOCKER LABELS

Key value pairs Key Add key Value Add value


The configuration details for App Mesh proxy. These fields are auto-configured for you after applying the App Mesh integration options above, otherwise must be configured manually. Learn More

Enable proxy configuration ☐

Log router integration

FireLens for Amazon ECS helps you route logs to an AWS service or AWS Partner Network (APN) destination for log storage and analysis. FireLens works with Fluentd and Fluent Bit. To auto-configure a log router container, complete the following fields and then choose **Apply**. Learn more

Enable FireLens integration ☐

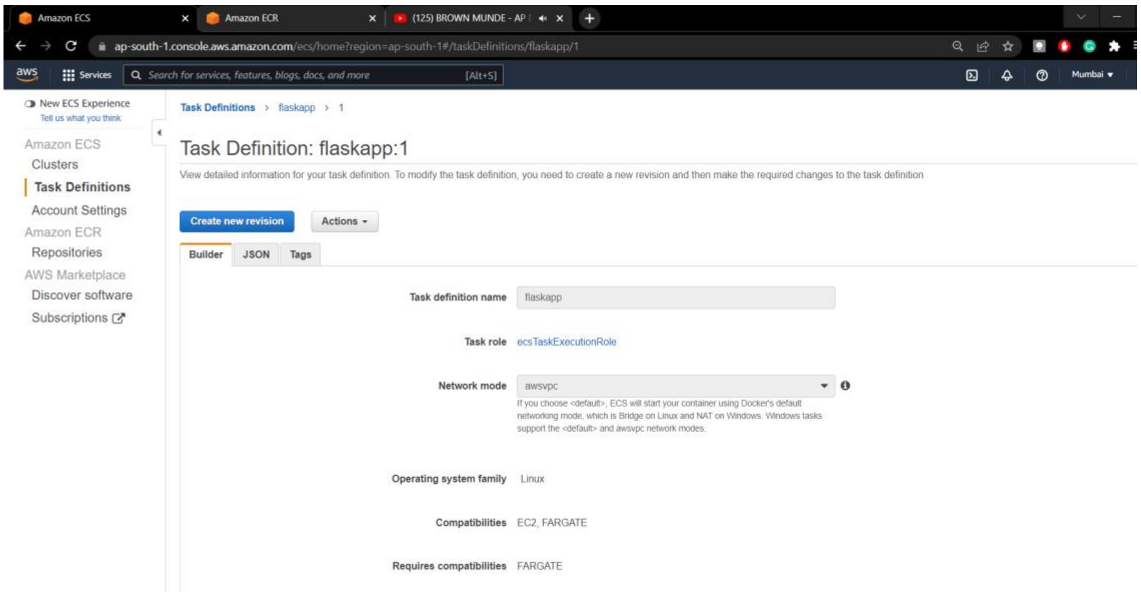
Volumes

Use a volume configuration to add volumes for use by the containers within a task. To add a volume, choose **Add volume**, complete the fields, and then choose **Add**. Learn more

Tags

*Required

We have our Task definition ready to deploy

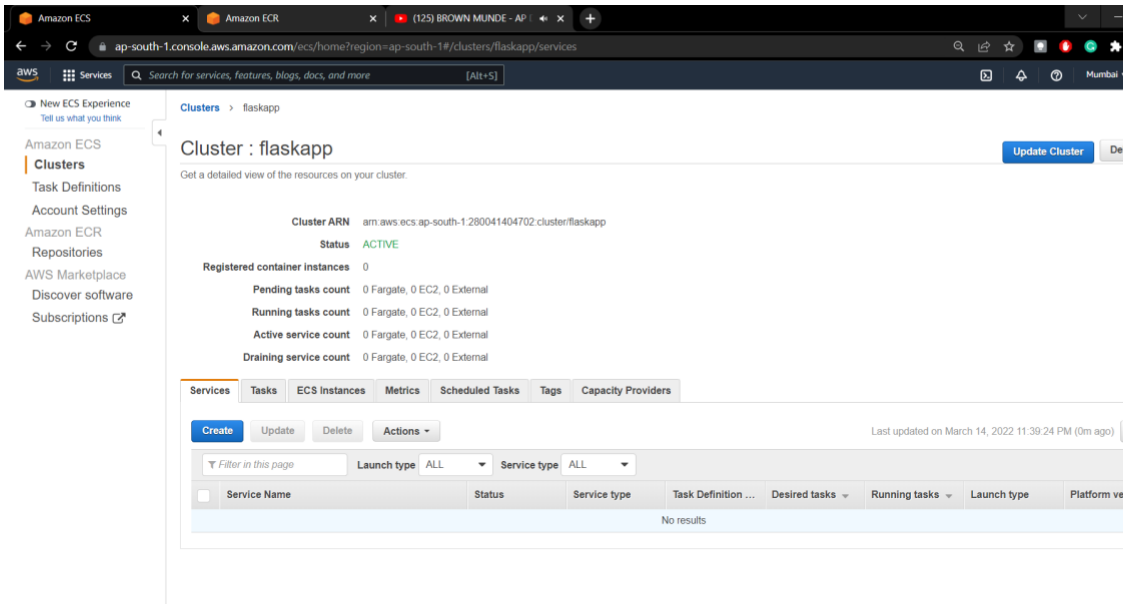

Task Definition: flaskapp:1

View detailed information for your task definition. To modify the task definition, you need to create a new revision and then make the required changes to the task definition

Task definition name	flaskapp
Task role	ecsTaskExecutionRole
Network mode	awsvpc
Operating system family	Linux
Compatibilities	EC2, FARGATE
Requires compatibilities	FARGATE

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. Windows tasks support the <default> and awsvpc network modes.

We can now navigate to our cluster



Clusters > flaskapp

Cluster : flaskapp

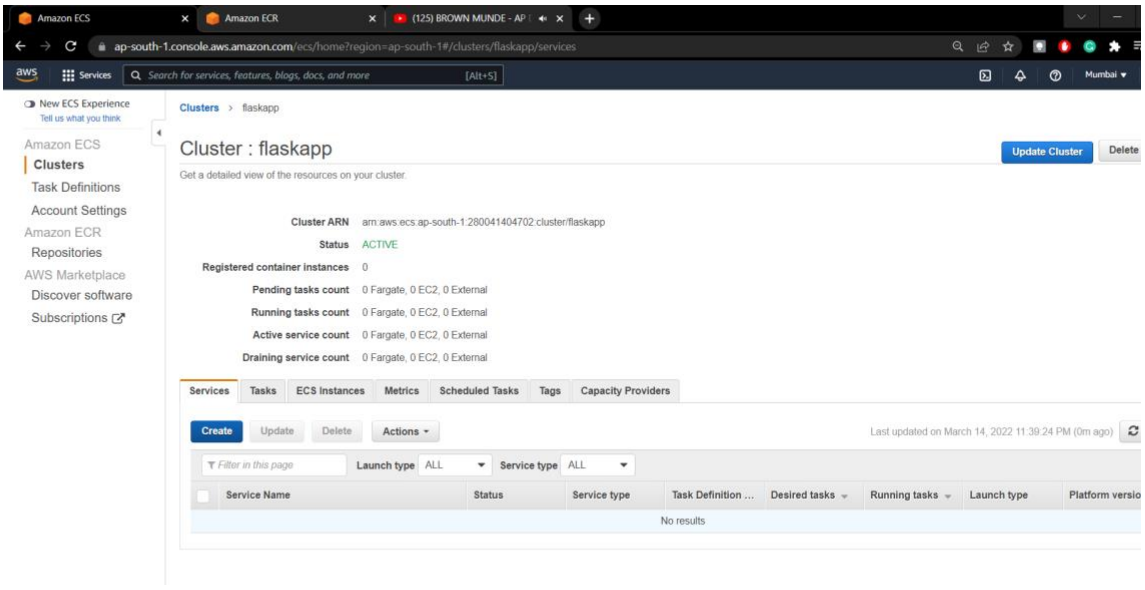
Get a detailed view of the resources on your cluster.

Cluster ARN	arn:aws:ecs:ap-south-1:280041404702:cluster/flaskapp
Status	ACTIVE
Registered container instances	0
Pending tasks count	0 Fargate, 0 EC2, 0 External
Running tasks count	0 Fargate, 0 EC2, 0 External
Active service count	0 Fargate, 0 EC2, 0 External
Draining service count	0 Fargate, 0 EC2, 0 External

Last updated on March 14, 2022 11:39:24 PM (0m ago)

Service Name	Status	Service type	Task Definition ...	Desired tasks	Running tasks	Launch type	Platform ve
No results							

We are now ready to create the service



Clusters > flaskapp

Cluster : flaskapp

Get a detailed view of the resources on your cluster.

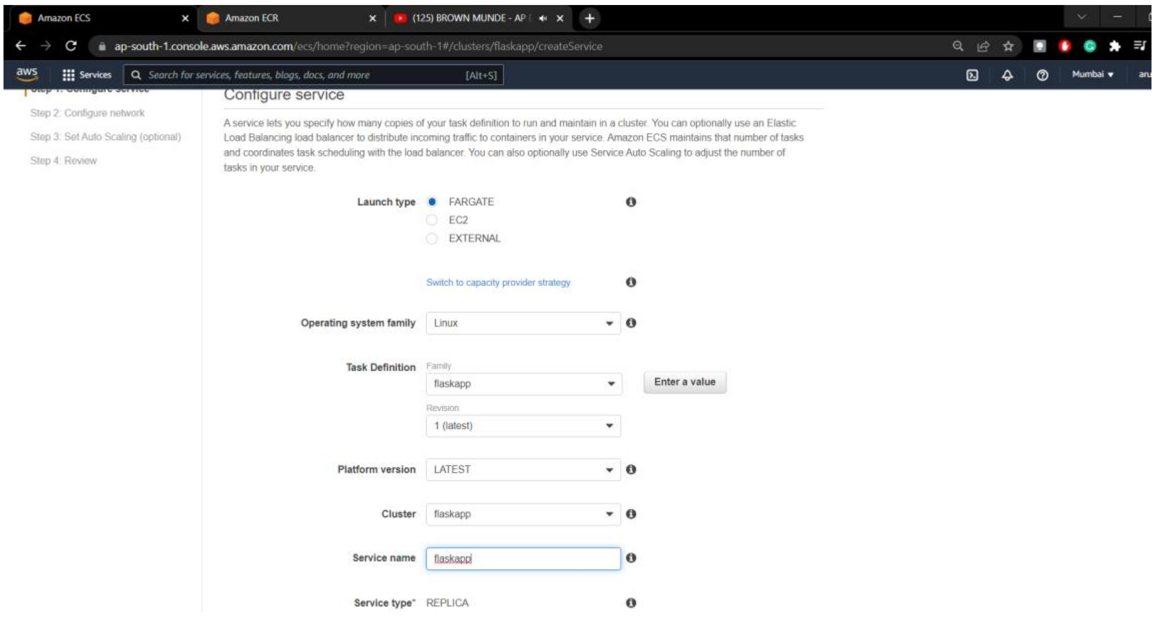
Cluster ARN	arn:aws:ecs:ap-south-1:280041404702:cluster/flaskapp
Status	ACTIVE
Registered container instances	0
Pending tasks count	0 Fargate, 0 EC2, 0 External
Running tasks count	0 Fargate, 0 EC2, 0 External
Active service count	0 Fargate, 0 EC2, 0 External
Draining service count	0 Fargate, 0 EC2, 0 External

Last updated on March 14, 2022 11:39:24 PM (0m ago)

Service Name	Status	Service type	Task Definition ...	Desired tasks	Running tasks	Launch type	Platform versio
No results							

Service configuration;

We have used fargate


Amazon ECS
Amazon ECR
(125) BROWN MUNDE - AP
ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters/flaskapp/createService
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Mumbai
Step 2: Configure network
Step 3: Set Auto Scaling (optional)
Step 4: Review

Configure service

A service lets you specify how many copies of your task definition to run and maintain in a cluster. You can optionally use an Elastic Load Balancing load balancer to distribute incoming traffic to containers in your service. Amazon ECS maintains that number of tasks and coordinates task scheduling with the load balancer. You can also optionally use Service Auto Scaling to adjust the number of tasks in your service.

Launch type: [✓ FARGATE] EC2 EXTERNAL

Switch to capacity provider strategy

Operating system family: Linux

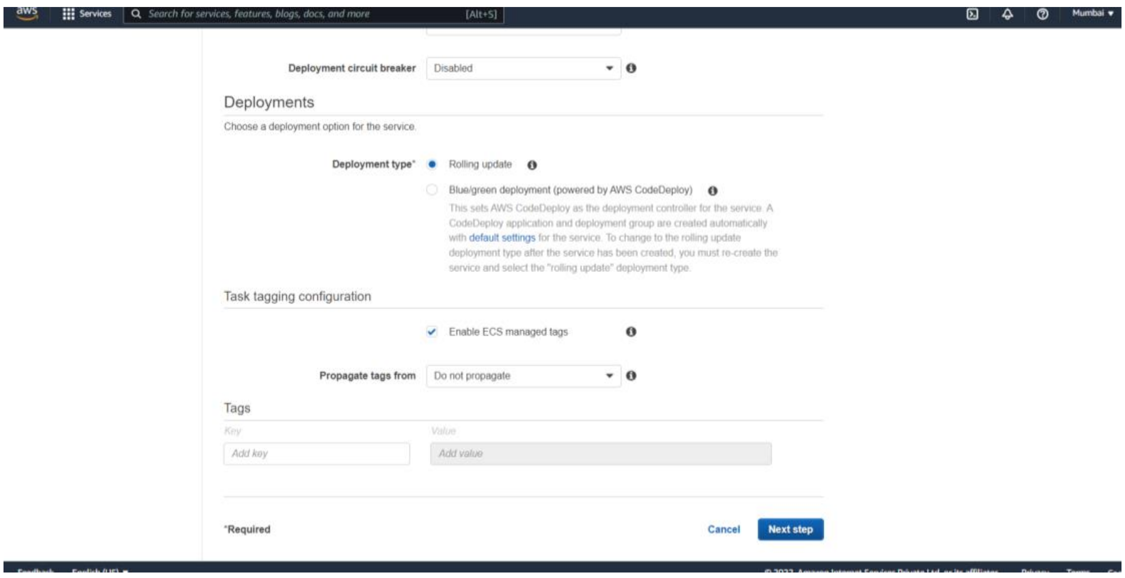
Task Definition: Family: flaskapp — Enter a value
Revision: 1 (latest)

Platform version: LATEST

Cluster: flaskapp

Service name: flaskapp

Service type*: REPLICA



Deployment circuit breaker: Disabled

Deployments

Choose a deployment option for the service.

Deployment type*: [✓ Rolling update]

Blue/green deployment (powered by AWS CodeDeploy)
This sets AWS CodeDeploy as the deployment controller for the service. A CodeDeploy application and deployment group are created automatically with default settings for the service. To change to the rolling update deployment type after the service has been created, you must re-create the service and select the "rolling update" deployment type.

Task tagging configuration

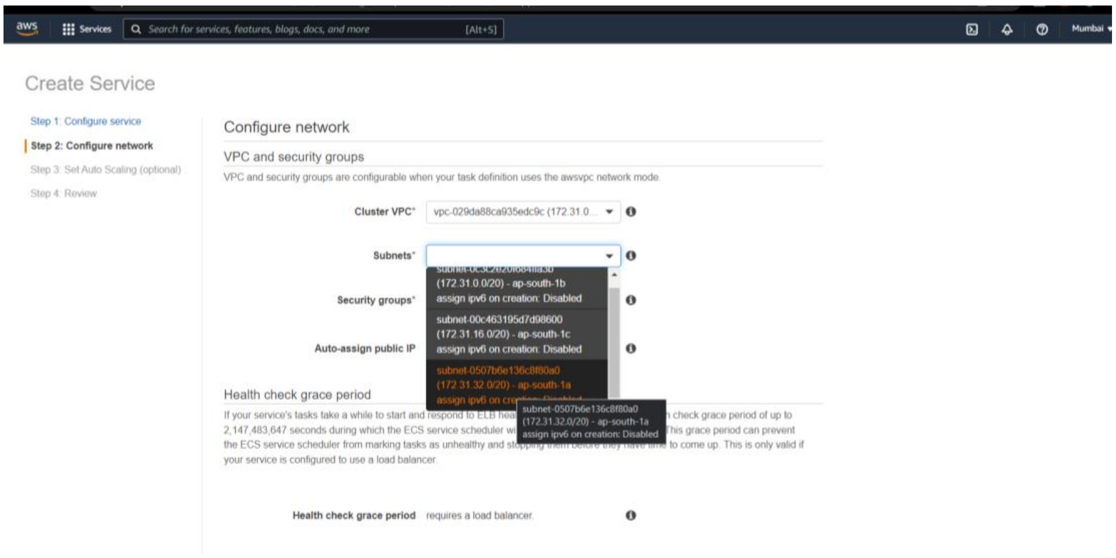
[✓] Enable ECS managed tags

Propagate tags from: Do not propagate

Tags

Key: Add key
Value: Add value

*Required



Create Service

Step 1: Configure service
Step 2: Configure network
Step 3: Set Auto Scaling (optional)
Step 4: Review

Configure network

VPC and security groups

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

Cluster VPC*: vpc-029da88ca935edc9c (172.31.0...

Subnets*:
(172.31.0.0/20) - ap-south-1b
assign ipv6 on creation: Disabled
subnet-00c463195d7d98600
(172.31.16.0/20) - ap-south-1c
assign ipv6 on creation: Disabled
subnet-0507b6e136c8f80a0
(172.31.32.0/20) - ap-south-1a
assign ipv6 on creation: Disabled

Security groups*

Auto-assign public IP

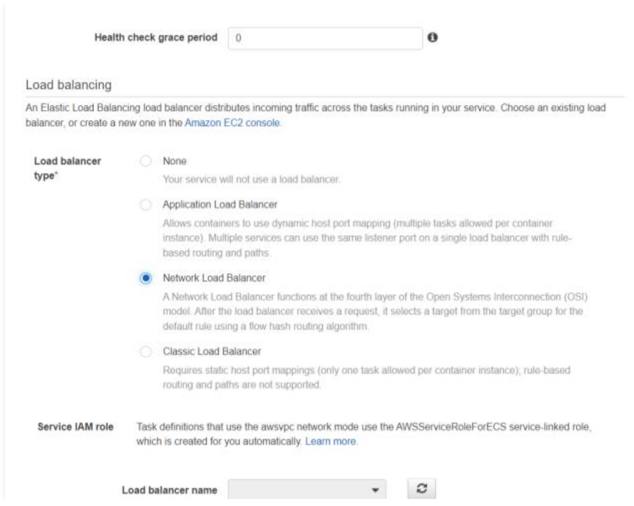
Health check grace period

If your service's tasks take a while to start and respond to ELB health checks, you can specify a health check grace period of up to 2,147,483,647 seconds during which the ECS service scheduler will ignore ELB health check status. This grace period can prevent the ECS service scheduler from marking tasks as unhealthy and stopping them before they have time to come up. This is only valid if your service is configured to use a load balancer.

Health check grace period: requires a load balancer.

Do the configurations as required

We can create the load balancer


Health check grace period: 0

Load balancing

An Elastic Load Balancing load balancer distributes incoming traffic across the tasks running in your service. Choose an existing load balancer, or create a new one in the Amazon EC2 console.

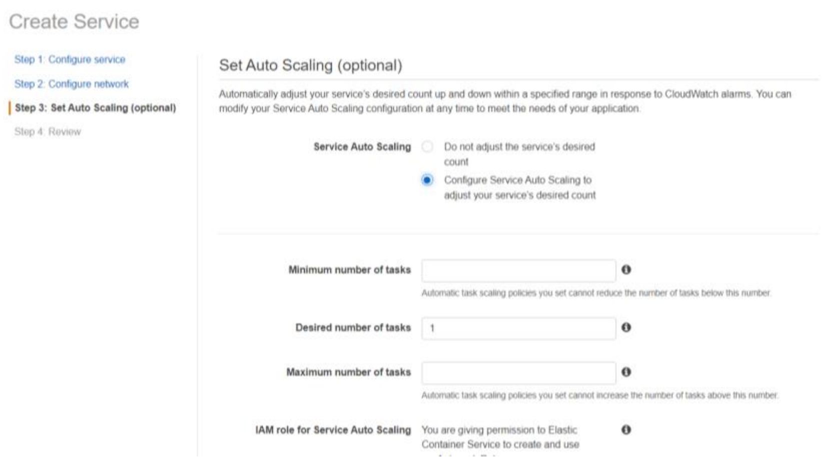
Load balancer type*

- None — Your service will not use a load balancer.
- Application Load Balancer — Allows containers to use dynamic host port mapping (multiple tasks allowed per container instance). Multiple services can use the same listener port on a single load balancer with rule-based routing and paths.
- [✓ Network Load Balancer] — A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it selects a target from the target group for the default rule using a flow hash routing algorithm.
- Classic Load Balancer — Requires static host port mappings (only one task allowed per container instance); rule-based routing and paths are not supported.

Service IAM role: Task definitions that use the awsvpc network mode use the AWSServiceRoleForECS service-linked role, which is created for you automatically. Learn more.

Load balancer name:

Auto configuration options are available as well


Create Service

Step 1: Configure service
Step 2: Configure network
Step 3: Set Auto Scaling (optional)
Step 4: Review

Set 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.

Service Auto Scaling:
- Do not adjust the service's desired count
- [✓ Configure Service Auto Scaling to adjust your service's desired count]

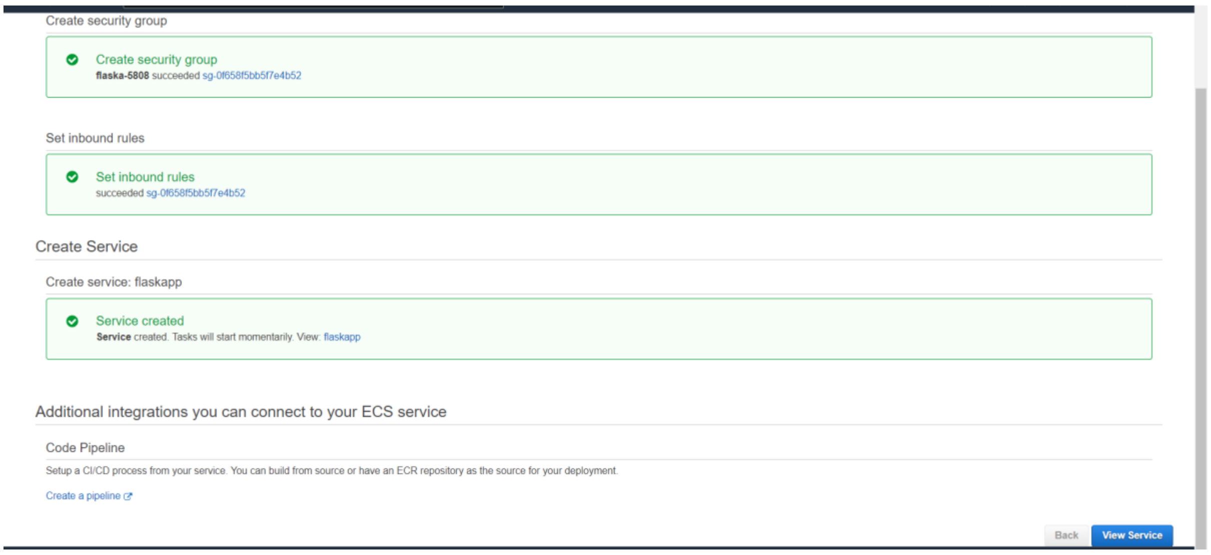
Minimum number of tasks:
Automatic task scaling policies you set cannot reduce the number of tasks below this number.

Desired number of tasks: 1

Maximum number of tasks:
Automatic task scaling policies you set cannot increase the number of tasks above this number.

IAM role for Service Auto Scaling: You are giving permission to Elastic Container Service to create and use

We have just configured the both auto scaling and load balancing options


Create security group

✓ Create security group
flaska-5808 succeeded sg-0f658f5bb5f7e4b52

Set inbound rules

✓ Set inbound rules
succeeded sg-0f658f5bb5f7e4b52

Create Service

Create service: flaskapp

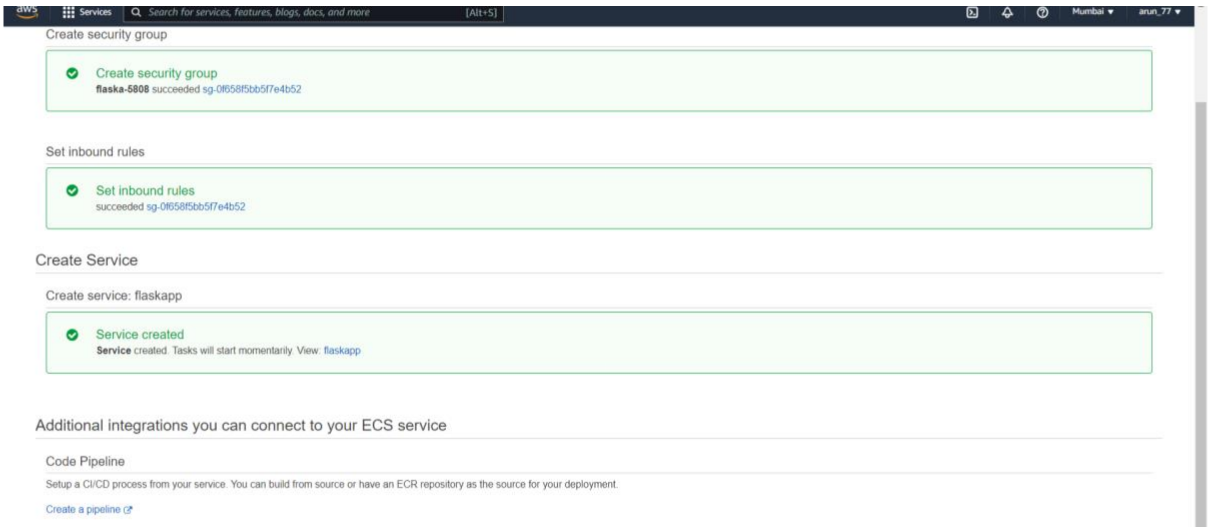
✓ Service created
Service created. Tasks will start momentarily. View: flaskapp

Additional integrations you can connect to your ECS service

Code Pipeline

Setup a CI/CD process from your service. You can build from source or have an ECR repository as the source for your deployment.

Create a pipeline



Create security group

Create security group
flaska-5808 succeeded sg-0f658f5bb5f7e4b52

Set inbound rules

Set inbound rules
succeeded sg-0f658f5bb5f7e4b52

Create Service

Create service: flaskapp

Service created
Service created. Tasks will start momentarily. View: flaskapp

Additional integrations you can connect to your ECS service

Code Pipeline

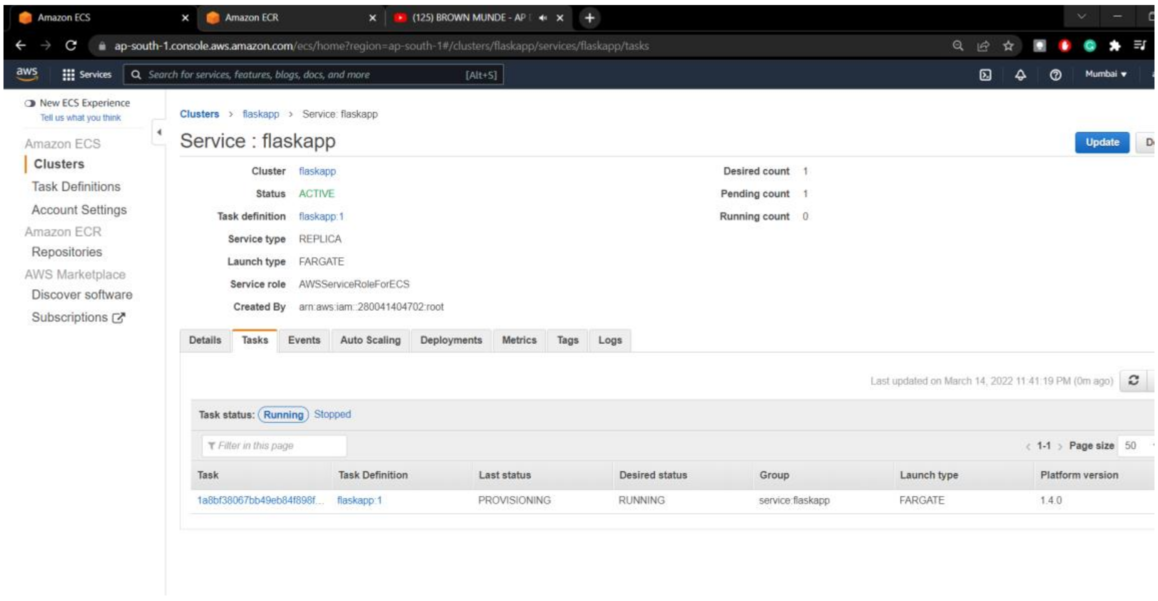
Setup a CI/CD process from your service. You can build from source or have an ECR repository as the source for your deployment.

Create a pipeline

The service is up and running

We can navigate to view our service and the cluster

We just need a couple of minutes



ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters/flaskapp/services/flaskapp/tasks

Clusters > flaskapp > Service: flaskapp

Service : flaskapp

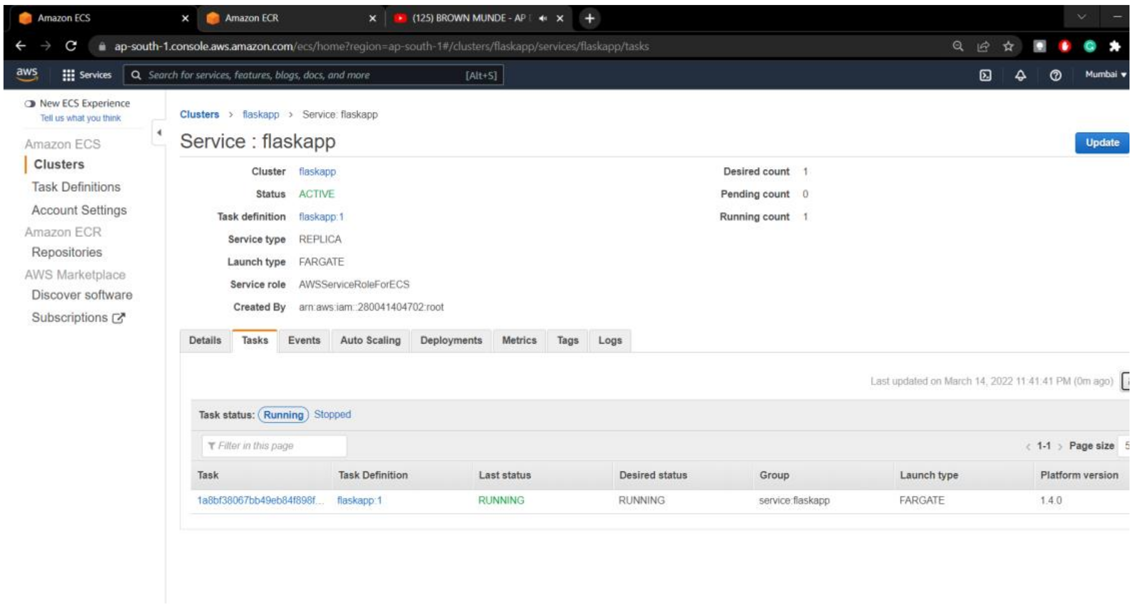
Cluster	flaskapp	**Desired count**	1
Status	ACTIVE	**Pending count**	1
Task definition	flaskapp:1	**Running count**	0
Service type	REPLICA		
Launch type	FARGATE		
Service role	AWSServiceRoleForECS		
Created By	arn:aws:iam::280041404702:root		

Last updated on March 14, 2022 11:41:19 PM (0m ago)

Task status: Running Stopped

Task	Task Definition	Last status	Desired status	Group	Launch type	Platform version
1a8bf38067bb49eb84f898f...	flaskapp:1	PROVISIONING	RUNNING	service:flaskapp	FARGATE	1.4.0

The service is ready to run and is running


ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters/flaskapp/services/flaskapp/tasks

Clusters > flaskapp > Service: flaskapp

Service : flaskapp

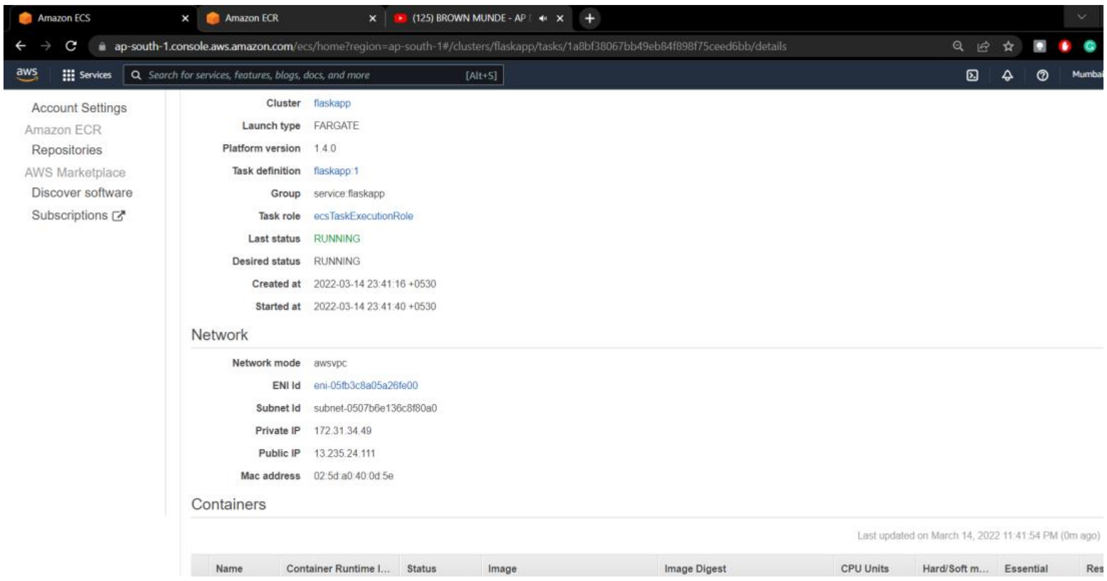
Cluster	flaskapp	Desired count	1
Status	ACTIVE	Pending count	0
Task definition	flaskapp:1	Running count	1
Service type	REPLICA		
Launch type	FARGATE		
Service role	AWSServiceRoleForECS		
Created By	arn:aws:iam::280041404702:root		

Last updated on March 14, 2022 11:41:41 PM (0m ago)

Task status: Running Stopped

Task	Task Definition	Last status	Desired status	Group	Launch type	Platform version
1a8bf38067bb49eb84f898f...	flaskapp:1	RUNNING	RUNNING	service:flaskapp	FARGATE	1.4.0

Just navigated to the service and copy the ip address (public)


ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters/flaskapp/tasks/1a8bf38067bb49eb84f898f75ceed6bb/details

Cluster	flaskapp
Launch type	FARGATE
Platform version	1.4.0
Task definition	flaskapp:1
Group	service:flaskapp
Task role	ecsTaskExecutionRole
Last status	RUNNING
Desired status	RUNNING
Created at	2022-03-14 23:41:16 +0530
Started at	2022-03-14 23:41:40 +0530

Network

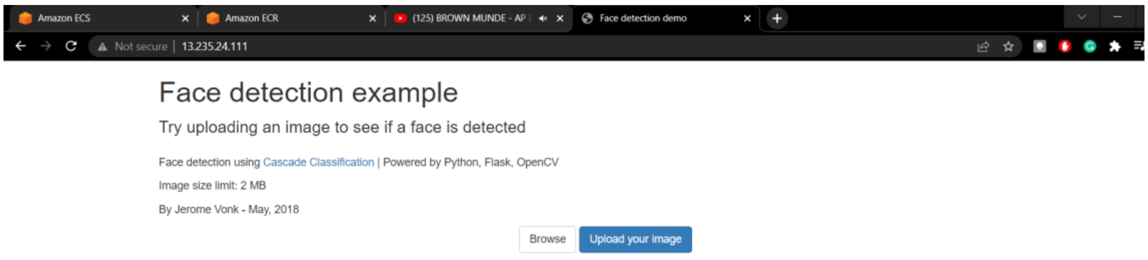
Network mode	awsvpc
ENI Id	eni-05fb3c8a05a26fe00
Subnet Id	subnet-0507b6e136c8f80a0
Private IP	172.31.34.49
Public IP	13.235.24.111
Mac address	02:5d:a0:40:0d:5e

Containers

Last updated on March 14, 2022 11:41:54 PM (0m ago)

Name	Container Runtime I...	Status	Image	Image Digest	CPU Units	Hard/Soft m...	Essential	Res

Search the ip address in the browser and you will get to see our website up and running


Not secure | 13.235.24.111

Face detection example

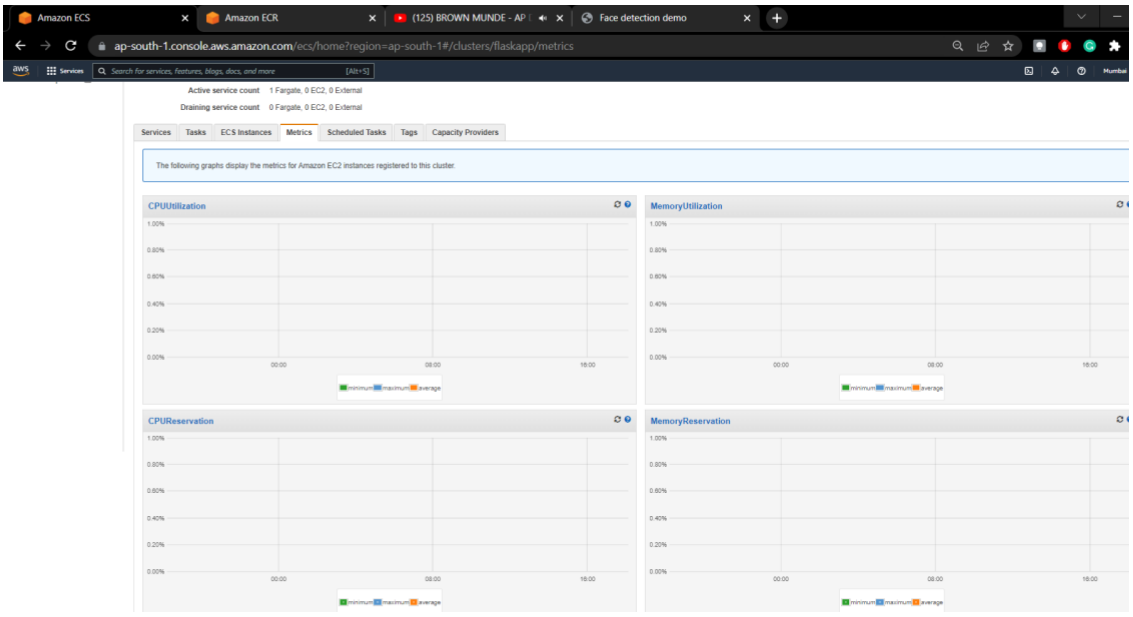
Try uploading an image to see if a face is detected

Face detection using Cascade Classification | Powered by Python, Flask, OpenCV

Image size limit: 2 MB

By Jerome Vonk - May, 2018

This is the matrixes that are provided by AWS, we can see the trends.



ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters/flaskapp/metrics

Active service count 1 Fargate, 0 EC2, 0 External

Draining service count 0 Fargate, 0 EC2, 0 External

The following graphs display the metrics for Amazon EC2 instances registered to this cluster.

We can also check the status, and insights of load balancing and auto scaling

We have tested on various platforms and the application was running and working seamlessly without any delay and it is very faster than the local machine because, the cluster works faster than that of a single machine.

Hence the project went successful