

Running a Docker Image on AWS

using AWS ECR and ECS

Pre-Requisites

- Docker should be installed on your system: - To install it follow the [LINK](#).
- You should have AWS CLI on your system and an AWS IAM user created and logged into AWS CLI using that user credentials. To do the same follow the steps: -
 - Download and Install AWS CLI: - [LINK](#)
 - Create IAM user with Administrator policy: - [LINK](#).
 - Config AWS CLI with AWS IAM credentials: -
Command: - “aws configure”
- Create a react application and build it.
- Now create a Dockerfile with the following content: -

```
FROM nginx:alpine
COPY build /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

- If you wish you can build the image now or later in the experiment.

Part 1: - Upload Image on ECR

1. Search for a service named: - “Elastic Container Registry”

The screenshot shows the AWS Management Console search results for 'ecr'. The search bar at the top contains 'ecr'. The results are categorized into 'Services' and 'Features'.

- Services:**
 - Elastic Container Registry: Fully-managed Docker container registry : Share and deploy container software, publ...
 - Secrets Manager: Easily rotate, manage, and retrieve secrets throughout their lifecycle
 - Key Management Service: Securely Generate and Manage AWS Encryption Keys
 - Systems Manager: AWS Systems Manager is a Central Place to View and Manage AWS Resources
- Features:**
 - Private registry: Elastic Container Registry feature
 - Repositories: Elastic Container Registry feature

On the right side of the console, there is a sidebar with sections like 'AWS', 'Getting started with AWS', 'Find certification', and 'New with AWS?'. The bottom of the screen includes standard AWS navigation links: CloudShell, Feedback, Language, Privacy, Terms, and Cookie preferences.

The screenshot shows the 'Amazon ECR > Repositories' page. The left sidebar has options for 'Private registry', 'Public registry', and 'Repositories'. Under 'Repositories', it lists 'Private repositories' and 'Public repositories'. The 'Private repositories' section shows a table with one row: 'No repositories'. A message below the table states 'No repositories were found'. At the top of the main content area, there are buttons for 'View push commands', 'Delete', 'Actions', and 'Create repository'. The bottom of the screen includes standard AWS navigation links: CloudShell, Feedback, Language, Privacy, Terms, and Cookie preferences.

2. Now click on “Create Repository”.

Amazon ECS Amazon ECS Elastic Container Registry - Create repository

ap-south-1.console.aws.amazon.com/ecr/create-repository?region=ap-south-1

Services Search [Alt+S]

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.

trial-dkr.ecr.ap-south-1.amazonaws.com/

0 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.

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

3. Now name your repository and keep rest of the settings on default and click “Create Repository”.

Amazon ECS Amazon ECS Elastic Container Registry - Create repository

ap-south-1.console.aws.amazon.com/ecr/create-repository?region=ap-south-1

Services Search [Alt+S]

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.

trial-ecr

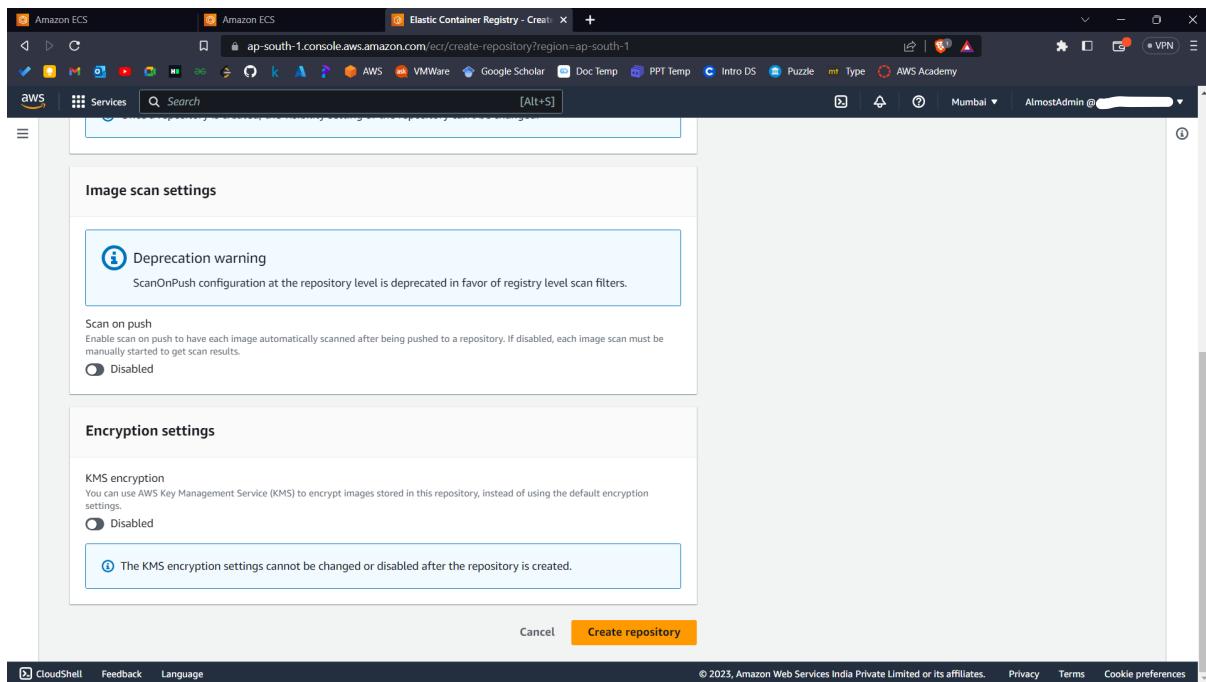
9 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

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.

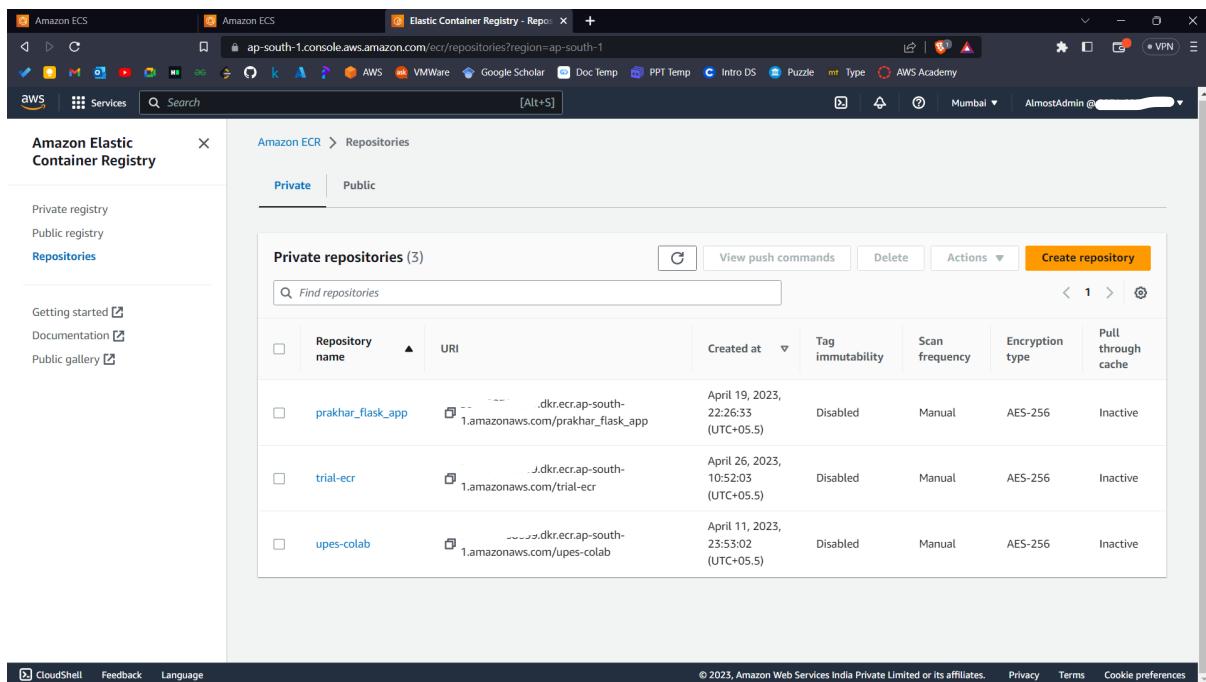
Disabled

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

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



4. Now click and open the currently created repo from the list of repositories.



5. Now click on view push commands.

The screenshot shows the AWS ECR console with the URL <https://ap-south-1.console.aws.amazon.com/ecr/repositories/private/trial-ecr?region=ap-south-1>. The left sidebar shows navigation options like 'Private registry', 'Public registry', 'Repositories', 'Summary', 'Images' (which is selected), 'Permissions', 'Lifecycle Policy', and 'Repository tags'. The main content area is titled 'trial-ecr' and shows the 'Images (0)' section. It includes a search bar and a table with columns: Image tag, Artifact type, Pushed at, Size (MB), Image URI, Digest, Scan status, and Vulnerabilities. A message 'No images' is displayed. At the top right of the main content area, there are buttons for 'View push commands', 'Edit', and other actions. The bottom of the screen shows standard AWS footer links: CloudShell, Feedback, Language, © 2023, Amazon Web Services India Private Limited or its affiliates., Privacy, Terms, and Cookie preferences.

You can use these commands to push your docker image to ECR.

The screenshot shows the 'Push commands for trial-ecr' dialog box. It has tabs for 'macOS / Linux' (selected) and 'Windows'. The dialog contains the following steps:

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  
.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.

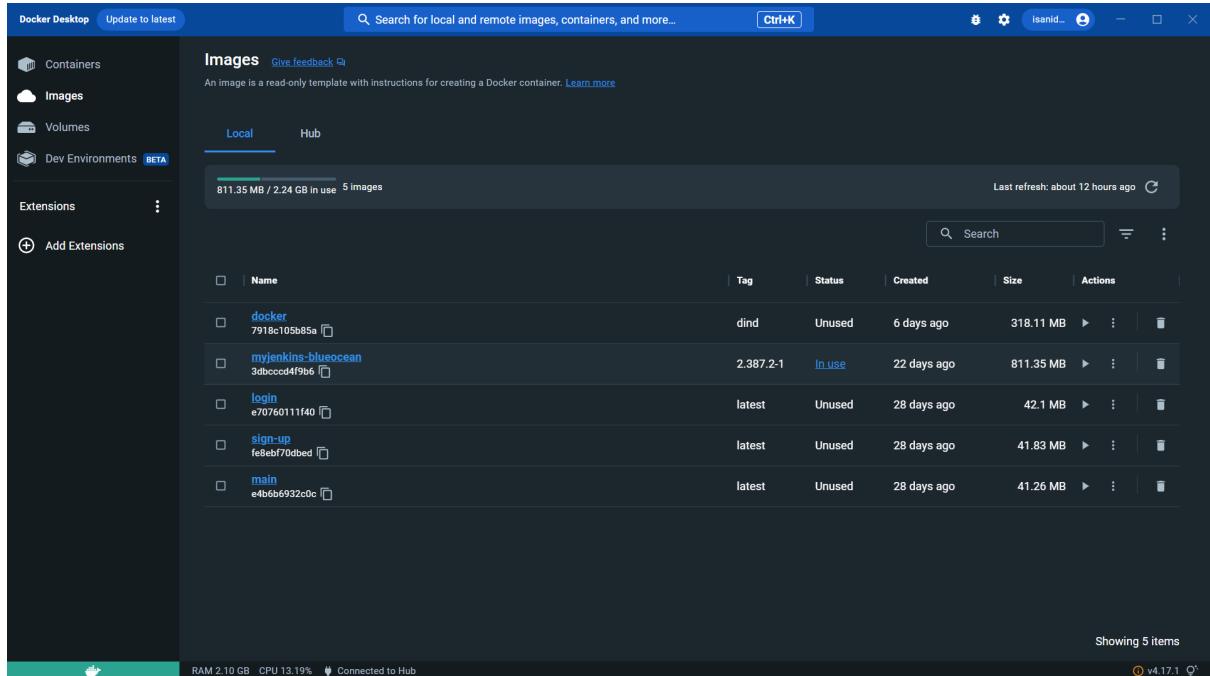
```
docker build -t trial-ecr .
```
3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag trial-ecr:latest .dkr.ecr.ap-south-1.amazonaws.com/trial-ecr:latest
```
4. Run the following command to push this image to your newly created AWS repository:

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

At the bottom right of the dialog is a 'Close' button.

6. Now firstly launch docker desktop on your system once to start the docker engine.



7. Now open terminal on your system and run the commands as per the instructions.

To list the docker images on your system: - “docker images”

A screenshot of a PowerShell window. The title bar says 'PowerShell'. The command 'docker images' is run, and the output is displayed in a table:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker	dind	7918c105b85a	5 days ago	318MB
myjenkins-blueocean	2.387.2-1	3dbcccd4f9b6	3 weeks ago	811MB
login	latest	e70760111f40	3 weeks ago	42.1MB
sign-up	latest	fe8ebf70dbed	4 weeks ago	41.8MB
main	latest	e4b6b6932c0c	4 weeks ago	41.3MB

Now decide which image you want to push then run the ECR push commands provided by AWS with a bit of manipulations as mentioned.

Commands you'll get from AWS: -

Push commands for trial-eqr

[macOS / Linux](#) | [Windows](#)

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

```
docker tag trial-eqr:latest 'kr.ecr.ap-south-1.amazonaws.com/trial-eqr:latest'
```
4. Run the following command to push this image to your newly created AWS repository:

```
docker push dkr.ecr.ap-south-1.amazonaws.com/trial-eqr:latest
```

[Close](#)

8. Run commands in terminal as per the instructions.

i) Command: -

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

Instructions: - Replace “<Account Id>” with your AWS account Id on top right corner of AWS console.

You can skip this command if the docker image already exists.

ii) Command: -

```
docker build -t <Image Name> .
```

Instructions: -

In terminal go inside the directory where the Dockerfile exists.

Replace "<Image Name>" with the name you want to give to your image and run the command and wait for it to build the image.

iii) Command: -

```
docker tag <Image Name>:latest <Account Id>.dkr.ecr.ap-south-1.amazonaws.com/<Repo Name>:<Image Name>
```

Instructions: -

Replace "<Image Name>" with the name you gave to your image.

Replace "<Account Id>" with your account id.

Replace "<Repo Name>" with your repository name.

You can also copy the command from AWS directly and do the respective manipulations.

iv) Command: -

```
docker push <Account Id>.dkr.ecr.ap-south-1.amazonaws.com/<Repo Name>:<Image Name>
```

Instructions: -

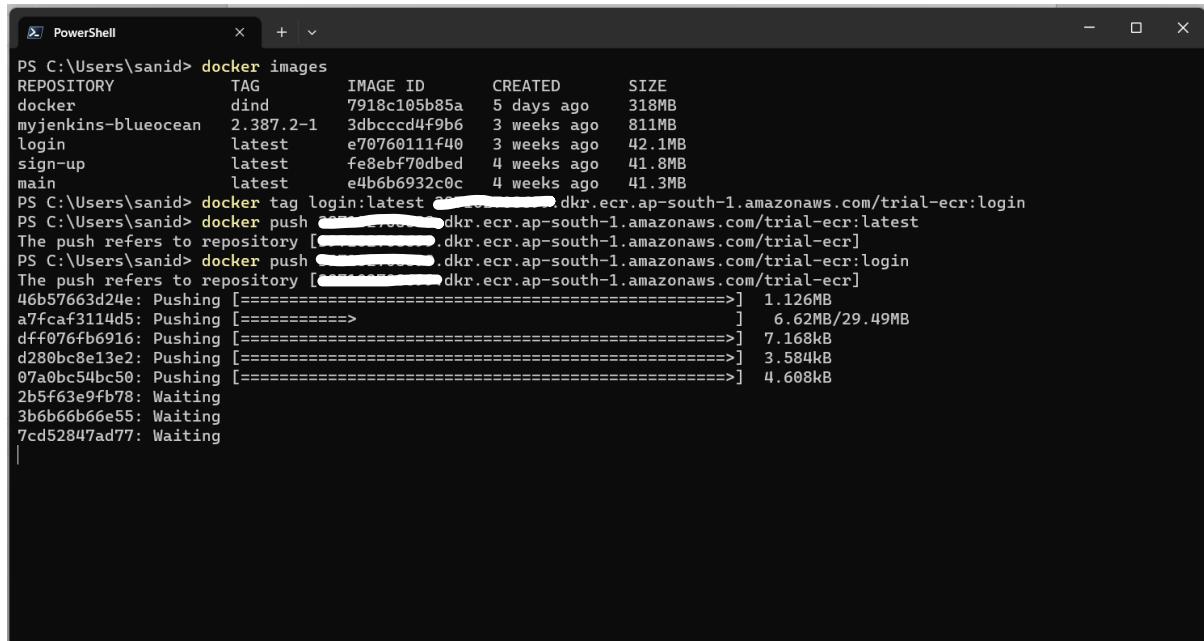
Replace "<Image Name>" with the name you gave to your image.

Replace "<Account Id>" with your account id.

Replace "<Repo Name>" with your repository name.

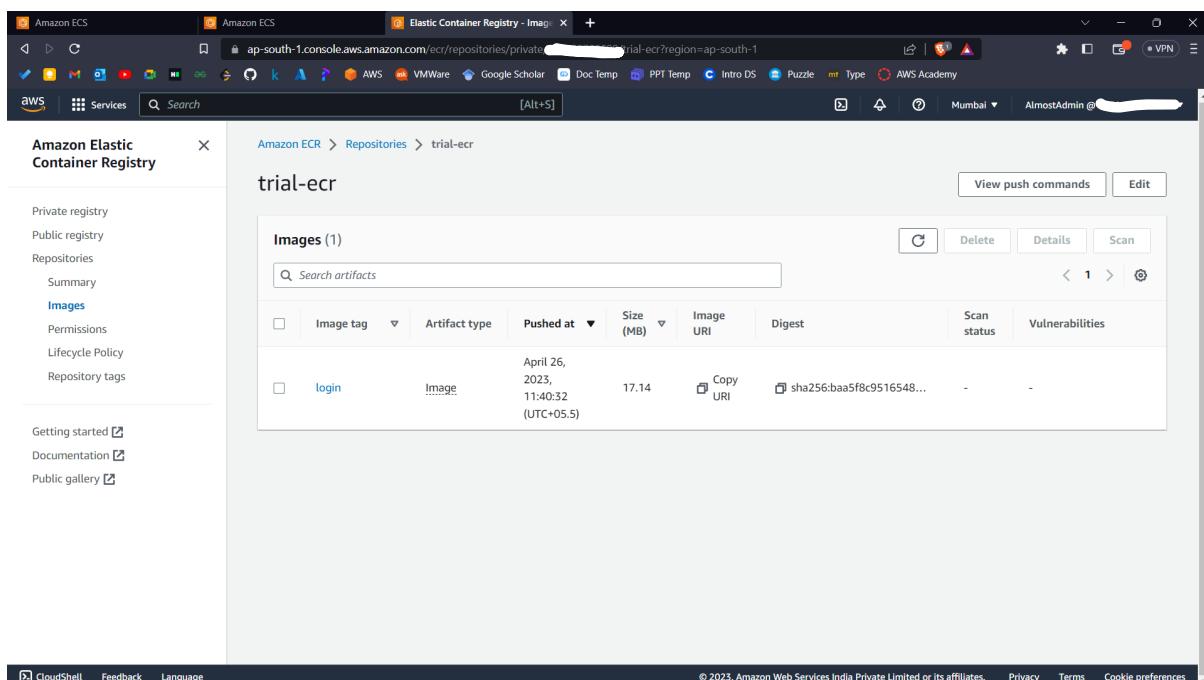
You can also copy the command from AWS directly and do the respective manipulations.

Your image is now successfully pushed on ECR.



```
PS C:\Users\sanid> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
docker          dind     7918c105b85a  5 days ago   318MB
myjenkins-blueocean  2.387.2-1  3dbcccd4f9b6  3 weeks ago  811MB
login           latest    e70760111f40  3 weeks ago  42.1MB
sign-up         latest    fe8ebf70bed   4 weeks ago  41.8MB
main            latest    e4b6b6932c0c   4 weeks ago  41.3MB
PS C:\Users\sanid> docker tag login:latest [REDACTED]dkr.ecr.ap-south-1.amazonaws.com/trial-ecr:login
PS C:\Users\sanid> docker push [REDACTED]dkr.ecr.ap-south-1.amazonaws.com/trial-ecr:latest
The push refers to repository [REDACTED].dkr.ecr.ap-south-1.amazonaws.com/trial-ecr]
PS C:\Users\sanid> docker push [REDACTED].dkr.ecr.ap-south-1.amazonaws.com/trial-ecr:login
The push refers to repository [REDACTED].dkr.ecr.ap-south-1.amazonaws.com/trial-ecr]
46b57663d24e: Pushing [=====>] 1.126MB
a7fcacf3114d5: Pushing [=====] 6.62MB/29.49MB
dff076fb6916: Pushing [=====>] 7.168kB
d280bc8e13e2: Pushing [=====>] 3.584kB
07a0bc54bc50: Pushing [=====>] 4.608kB
2b5f63e9fb78: Waiting
3b6b66b66e55: Waiting
7cd52847ad77: Waiting
|
```

Keep in mind Image URI is visible in over here.



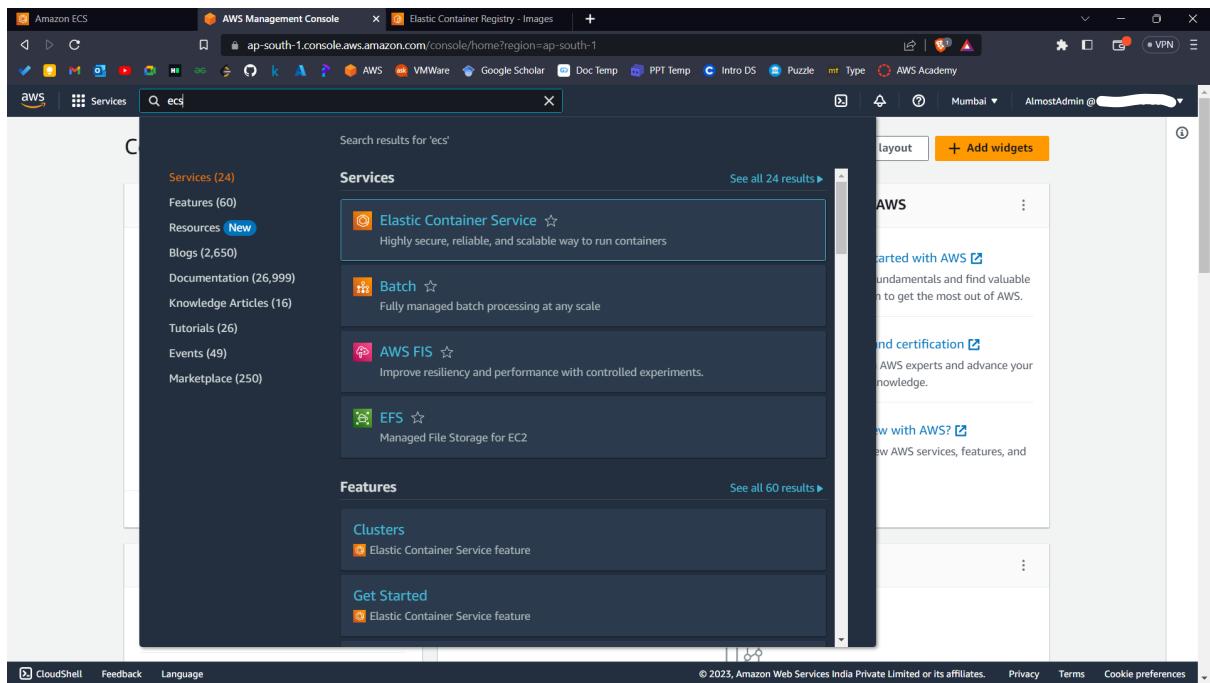
The screenshot shows the AWS CloudShell interface with the following details:

- Amazon ECS**: Two tabs are open, one for "Amazon ECS" and another for "Elastic Container Registry - Images".
- URL**: `ap-south-1.console.aws.amazon.com/ecr/repositories/private/[REDACTED]/trial-ecr?region=ap-south-1`
- Left Sidebar (Amazon Elastic Container Registry)**:
 - Private registry
 - Public registry
 - Repositories
 - Summary
 - Images** (selected)
 - Permissions
 - Lifecycle Policy
 - Repository tags
- Getting started**, **Documentation**, and **Public gallery** links are also present.
- Amazon ECR > Repositories > trial-ecr**: The main view shows the repository details.
- trial-ecr**: The repository name.
- Images (1)**: A table listing the image details.

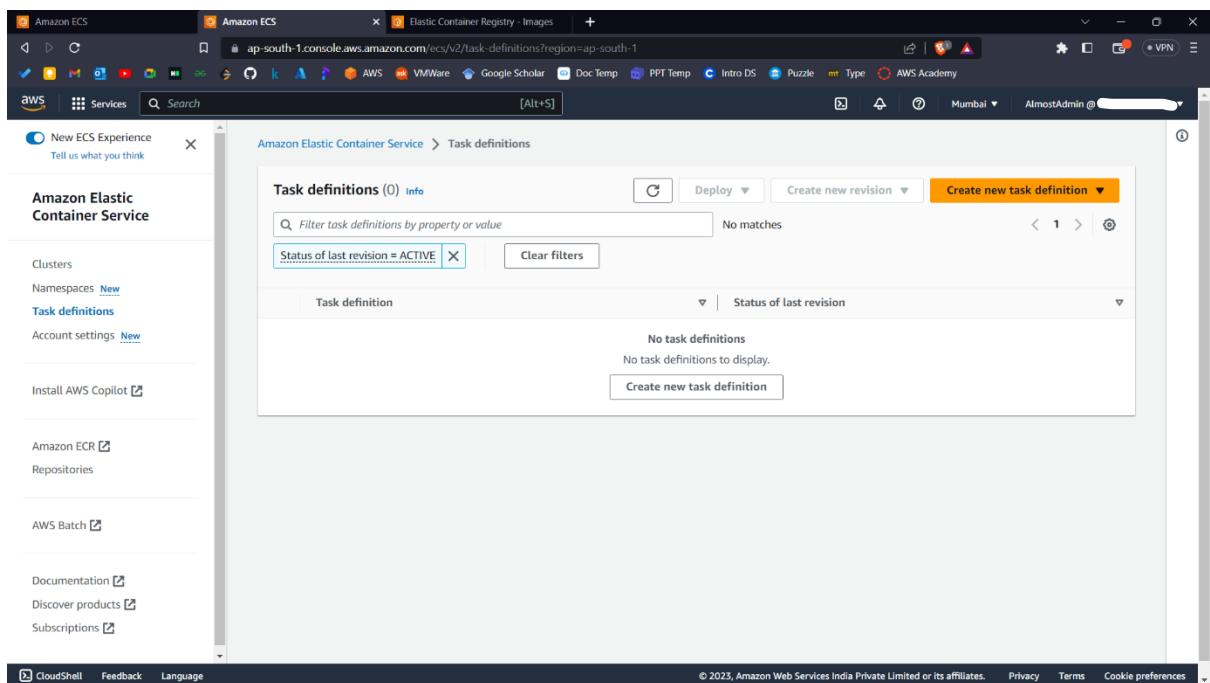
	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulnerabilities
<input type="checkbox"/>	login	Image	April 26, 2023, 11:40:32 (UTC+05:5)	17.14	Copy URI	sha256:baa5f8c9516548...	-	-
- View push commands** and **Edit** buttons are located at the top right of the image table.
- CloudShell**, **Feedback**, **Language**, **© 2023, Amazon Web Services India Private Limited or its affiliates.**, **Privacy**, **Terms**, and **Cookie preferences** are at the bottom.

Part 2: - Create a Task Definition

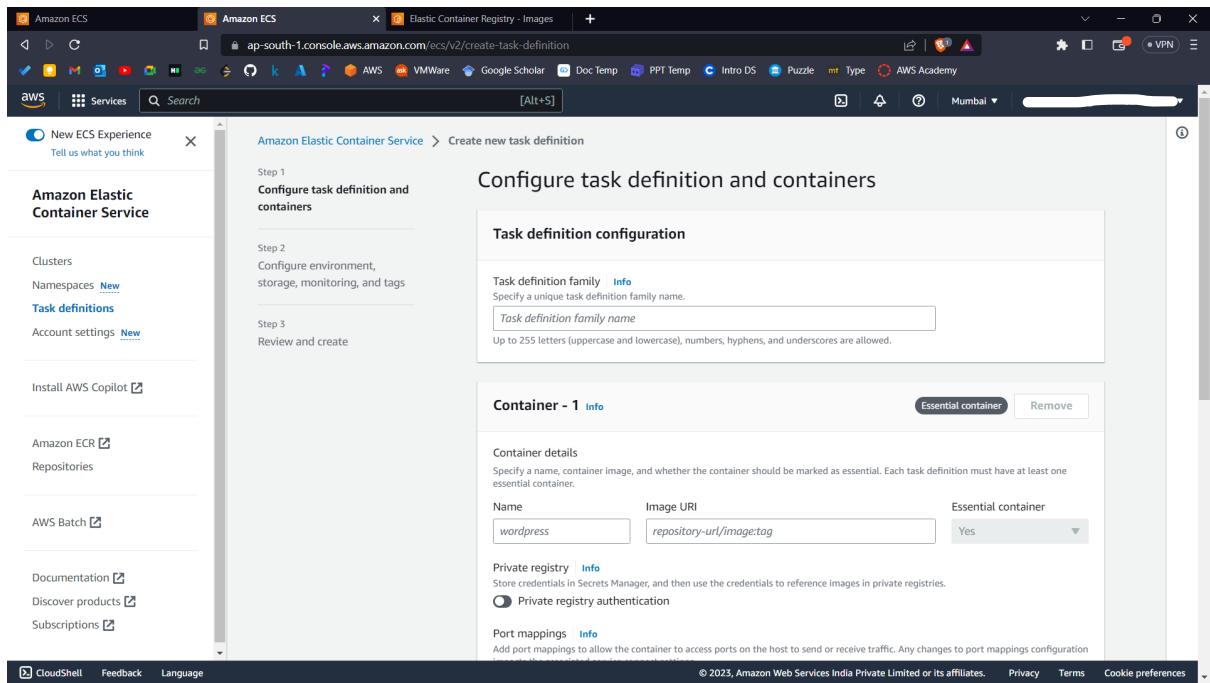
1. Go to AWS console and search Elastic Container Service.



2. Now go to task definitions in left side.



3. Click on Create New Task Definition and select Create New Task Definition.



4. Now name your task definition family.

5. Give your container 1 details: -

- Name
- Image URI – Go to AWS ECR inside your repository and copy the image URI of your image as mentioned before to keep in mind.
- In Container Port, set the port on which your docker image runs.
- And keep rest settings to default as shown in screenshots.

Configure task definition and containers

Task definition configuration

Task definition family [Info](#)
Specify a unique task definition family name.
Sani-Login-Task
Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

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
Sani-Login	dkr.ecr.ap-south-1.amazonaws.com/tria	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	sani-login-80-tcp	HTTP

Add more port mappings

Environment variables - optional [Info](#)

Add individually
Add a key-value pair to specify an environment variable.

Add environment variable

Add from file
Add environment variables in bulk by providing an environment file hosted on Amazon S3.

Add environment file

You can add 10 more environment files.

► HealthCheck - optional [Info](#)

► Docker configuration - optional [Info](#)

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	sani-login-80-tcp	HTTP

Add more port mappings

Environment variables - optional [Info](#)

Add individually
Add a key-value pair to specify an environment variable.

Add environment variable

Add from file
Add environment variables in bulk by providing an environment file hosted on Amazon S3.

Add environment file

You can add 10 more environment files.

► HealthCheck - optional [Info](#)

► Docker configuration - optional [Info](#)

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

6. Click next.

Now configure the environment.

App environment: - AWS Fargate (Default)

Operating System/Architecture: - Linux/X86_64

Task Size: -

- CPU: - .25 vCPU
- Memory: - .5 GB

Task Role: - None.

Task Execution Role: - ecsTaskExecutionRole. (For the first time it will be created automatically)

Leave below settings to default.

The screenshot shows the 'Create new task definition' wizard in the Amazon Elastic Container Service (ECS) console. The left sidebar shows 'Namespaces' and 'Task definitions' selected. The main pane is titled 'Configure environment, storage, monitoring, and tags'. Under the 'Environment' section, 'App environment' is set to 'AWS Fargate (serverless)', 'Operating system/Architecture' is set to 'Linux/X86_64', and 'Task size' is set to '.25 vCPU' and '.5 GB'. The 'Task execution role' dropdown is empty.

The screenshot shows the 'Create new task definition' wizard in the Amazon Elastic Container Service (ECS) console. The left sidebar shows 'Namespaces' and 'Task definitions' selected. The main pane is titled 'Configure environment, storage, monitoring, and tags'. Under the 'Task size' section, 'CPU' is set to '.25 vCPU' and 'Memory' is set to '.5 GB'. Under the 'Container size - optional' section, 'None' is selected. Under the 'Task roles, network mode - conditional' section, 'Task execution role' is set to 'ecsTaskExecutionRole' and 'Network mode' is set to 'awsvpc'. At the bottom right, there is a 'Storage - optional' section with an 'Add volume' button.

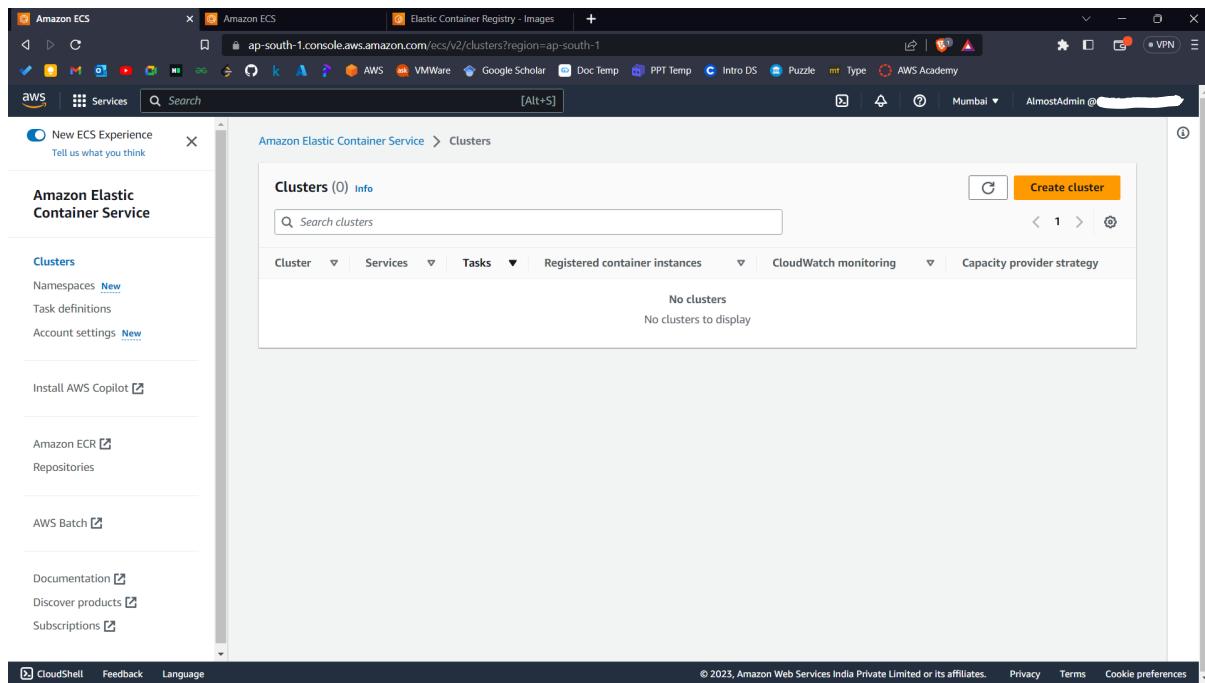
7. Click next.
8. Confirm the configurations and click create.

The screenshot shows the AWS Management Console interface for the Amazon Elastic Container Service (ECS). The left sidebar navigation bar includes links for Clusters, Namespaces (New), Task definitions (selected), Account settings (New), Install AWS Copilot, Amazon ECR, and AWS Batch. The main content area displays the 'Task definitions' section for the 'Sani-Login-Task'. A prominent green success message at the top states: 'Task definition successfully created' and 'Sani-Login-Task:1 has been successfully created. You can use this task definition to deploy a service or run a task.' Below this, the 'Overview' tab is selected, showing details such as ARN (arn:aws:ecs:ap-south-1:77102220:task-definition/Sani-Login-Task:1), Status (ACTIVE), Time created (4/26/2023, 07:00:29 UTC), App environment (FARGATE), Task role (-), Task execution role (ecsTaskExecutionRole), Operating system/Architecture (Linux/X86_64), and Network mode (awsvpc). Other tabs available are JSON, Storage, and Tags. At the bottom of the page, there are links for Documentation, Discover products, Subscriptions, CloudShell, Feedback, Language, and a footer with copyright information and links for Privacy, Terms, and Cookie preferences.

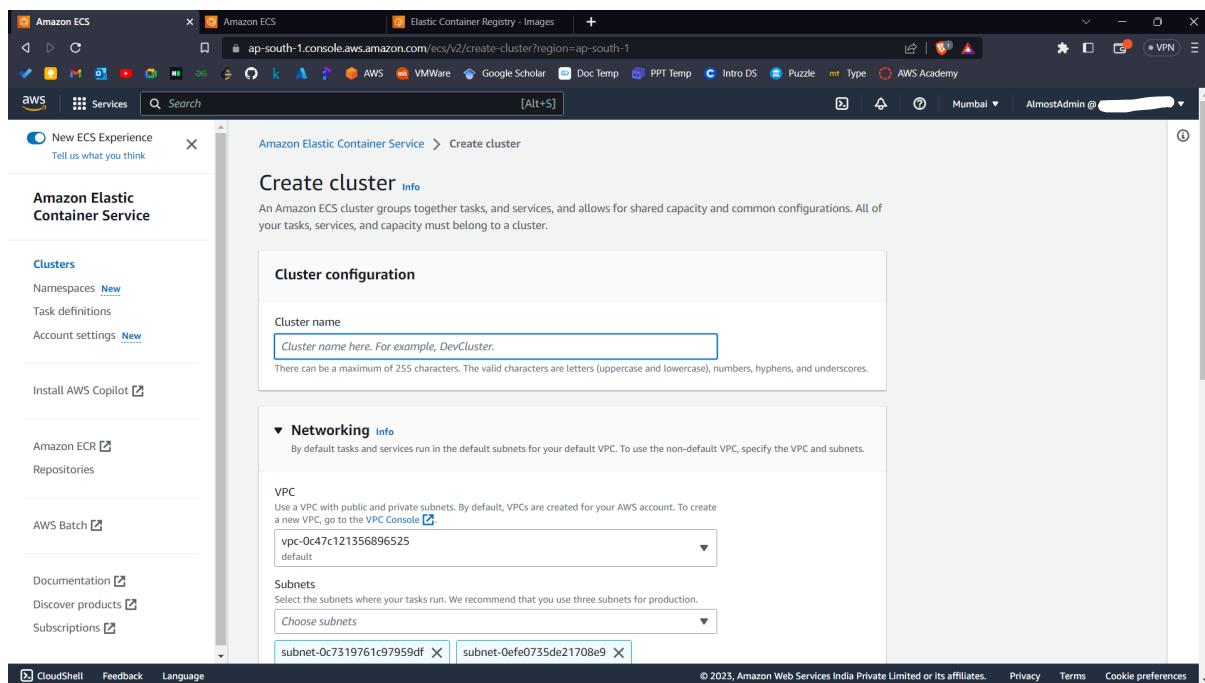
Your Task Definition is created successfully.

Part 3: - Create a Cluster and Run Task

1. In Elastic Container Service, go to clusters.



2. Click create cluster.



3. Give cluster name.

Keep rest things to default.

The screenshot shows the 'Networking' section of the AWS ECS 'Create Cluster' wizard. It includes fields for selecting a VPC (set to 'vpc-0c47c121356896525') and three subnets ('subnet-0c7319761c97959df', 'subnet-0e0fe0735de21708e9', and 'subnet-0755af44328619aaf'). A 'Default namespace - optional' field is present but empty. The 'Infrastructure' section below indicates the cluster is configured for AWS Fargate (serverless) with two capacity providers.

The screenshot shows the 'Infrastructure' section of the AWS ECS 'Create Cluster' wizard. It lists three subnets ('subnet-0c7319761c97959df', 'subnet-0e0fe0735de21708e9', and 'subnet-0755af44328619aaf'). A 'Default namespace - optional' field is empty. The 'AWS Fargate (serverless)' checkbox is selected. Below it, there are three other capacity provider options: 'Amazon EC2 instances' (unchecked), 'External instances using ECS Anywhere' (unchecked), and 'Monitoring - optional' (unchecked).

4. Click create.

The screenshot shows the AWS Elastic Container Service (ECS) Clusters page. On the left, there's a sidebar with links for New ECS Experience, Amazon Elastic Container Service (selected), Clusters (with sub-links for Namespaces, Task definitions, and Account settings), and other services like Amazon ECR and AWS Batch. The main content area shows a table for Clusters. It has one entry: "Sani-Cluster" with a status of "0" under Cluster, "No tasks running" under Services, and "0" under Tasks. To the right of the table is a "Create cluster" button. At the bottom of the page, there are links for CloudShell, Feedback, Language, and copyright information from 2023.

5. Now open the cluster.

The screenshot shows the AWS ECS Cluster overview page for "Sani-Cluster". The left sidebar is identical to the previous screenshot. The main content area is titled "Sani-Cluster" and contains a "Cluster overview" section. It shows the ARN "Sani-Cluster" with a status of "Active", CloudWatch monitoring set to "Default", and no registered container instances. Below this is a "Services" section showing one service named "Draining" with an active status, pending tasks, and running tasks. A "Tasks" section is also present. At the bottom, there's a "Services (0) Info" table with columns for Service name, Status, ARN, Service..., Deployments and tasks, and Last dep. The table shows "No services" and "No services to display". The bottom of the page includes standard AWS footer links for CloudShell, Feedback, Language, and copyright information from 2023.

6. Go to tasks.

The screenshot shows the AWS ECS Cluster overview page. On the left, there's a sidebar with options like 'Clusters', 'Namespaces', 'Task definitions', and 'Account settings'. The main area displays 'Cluster overview' with details for 'Sani-Cluster': ARN, Status (Active), CloudWatch monitoring (Default), and Registered container instances (None). Below this is a 'Tasks' section with a table showing one task: Draining, Active, Pending, and Running. At the bottom, there's a 'Tasks (0)' section with a 'Run new task' button.

7. Click Run New Task.

The screenshot shows the 'Run new task' configuration page for AWS Fargate. It includes sections for 'Environment' (Existing cluster: Sani-Cluster, Compute configuration (advanced) with Launch type set to Launch tasks directly without the use of a capacity provider strategy, and Launch type set to FARGATE), 'Platform version' (LATEST), and 'Deployment configuration'.

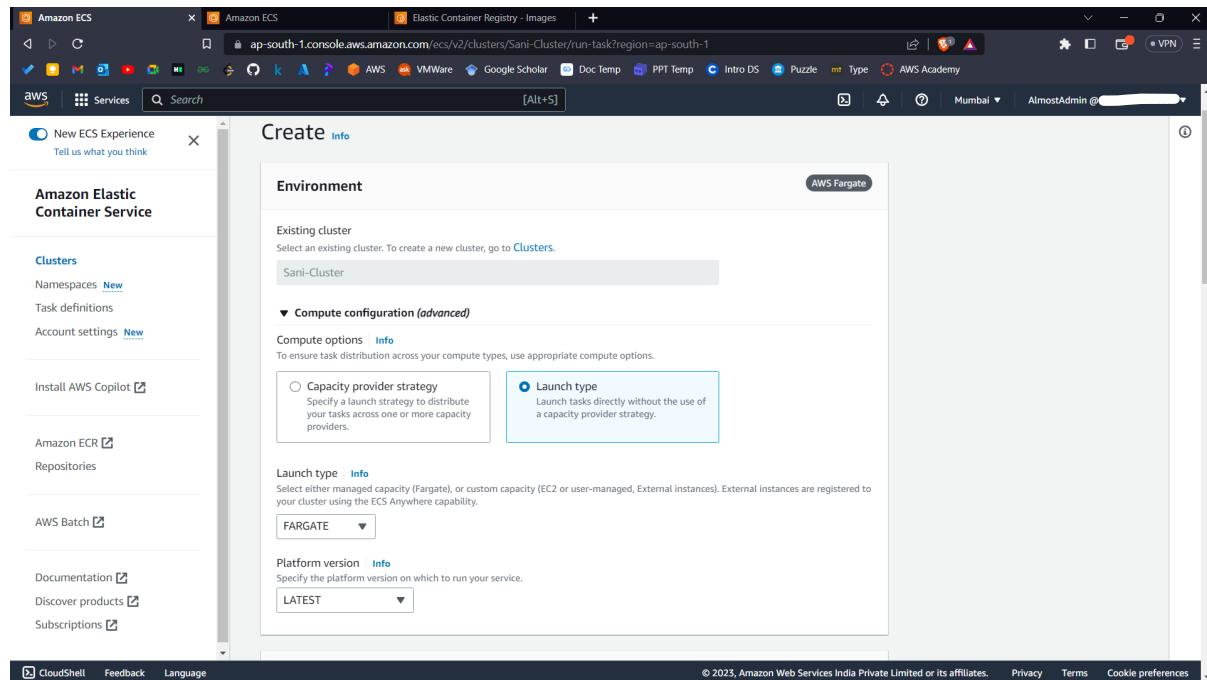
8. Configure according to the instructions: -

Environment: -

Compute options: - Launch Type.

Launch Type: - FARGATE

Platform Version: - LATEST.



Deployment Configuration: -

Application Type: - Task

Family: - Name of Task Definition you created

Revision: - 1(LATEST) (Default)

Desired Tasks: - 1

The screenshot shows the AWS ECS Deployment Configuration page. The left sidebar is for the Amazon Elastic Container Service, with 'Clusters' selected. The main content area is titled 'Deployment configuration'. Under 'Application type', 'Task' is selected. In the 'Task definition' section, 'Specify the revision manually' is checked, and 'Sani-Login-Task' is selected from the 'Family' dropdown with '1 (LATEST)' in the 'Revision' dropdown. The 'Desired tasks' field contains '1'. The 'Task group' field is empty.

Networking: -

Security Groups: -

Go to AWS console on new tab and search for EC2 and go to security groups.

The screenshot shows the AWS EC2 Management Console with the 'Security Groups' page open. The left sidebar navigation includes 'Launch templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images' (AMIs, AMI Catalog), 'Elastic Block Store' (Volumes, Snapshots, Lifecycle Manager), 'Network & Security' (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), 'Load Balancing' (Load Balancers, Target Groups), and 'CloudShell', 'Feedback', 'Language'. The main content area displays a table titled 'Security Groups (2) Info' with columns: Name, Security group ID, Security group name, VPC ID, Description, and Owner. Two entries are listed: 'sg-0625b02a97ff4f521' (ECS_Security, vpc-0c47c121356896525, Allow Public Access, 397102708699) and 'sg-099042c4546c7c99f' (default, vpc-0c47c121356896525, default VPC security gr..., 397102708699). Buttons for 'Actions', 'Export security groups to CSV', and 'Create security group' are at the top right of the table.

Create a new Security Group.

The screenshot shows the 'Create security group' wizard in the AWS EC2 Management Console. The URL in the address bar is 'ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateSecurityGroup'. The page title is 'Create security group info'. A sub-header says 'A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.' The 'Basic details' section contains fields for 'Security group name' (set to 'ESC-Security-Grp'), 'Description' (set to 'For ECS'), and 'VPC' (set to 'vpc-0c47c121356896525'). The 'Inbound rules' section has a table with columns: Type, Protocol, Port range, Source, and Description - optional. Under 'Type', 'All traffic' is selected. Under 'Protocol', 'All' is selected. Under 'Port range', 'Anywh...' is selected. Under 'Source', '0.0.0.0/0' is listed. Buttons for 'Delete' and 'Next Step' are at the bottom right of the 'Inbound rules' section.

Add an inbound and outbound rule with the following config.

Type: - All Traffic

Source: - Anywhere-IPv4

The image shows two screenshots of the AWS CloudFormation console. The top screenshot is titled 'Inbound rules' and shows a configuration for an AWS Lambda function. It has 'All traffic' selected for Type, 'All' for Protocol, and 'All' for Port range. The Source dropdown is set to 'Anywhere-IPv4'. The bottom screenshot is titled 'Outbound rules' and shows a similar configuration, also with 'All traffic' selected for Type, 'All' for Protocol, and 'All' for Port range. The Destination dropdown is set to 'Anywhere-IPv4'. Both screenshots include an 'Add rule' button and a 'Delete' button for each rule entry.

Now again go back to previous tab of ECS.

9. Select security group as the once you created just now. (This would work in further experiments also).
And now click Create.

The image shows the 'Task Definitions' section of the AWS ECS console. Under the 'Task definitions' tab, there is a 'Create new task definition' button. Below it, there are sections for 'Subnets' (listing 'subnet-0c7319761c97959df ap-south-1c' and 'subnet-0efe0735de21708e9 ap-south-1b'), 'Security group' (with 'Use an existing security group' selected), and 'Container overrides' and 'Task overrides' sections. The security group dropdown lists three options: 'sg-0625b02a97ff4f521 ECS_Security', 'sg-099042c4546c7c99f default', and 'sg-036aeba424e952d78 ESC-Security-Grp'. The 'sg-036aeba424e952d78' option is checked. At the bottom, there are 'Next Step' and 'Create' buttons.

The screenshot shows the AWS ECS console with the URL ap-south-1.console.aws.amazon.com/ecs/v2/clusters/Sani-Cluster/tasks?region=ap-south-1. The cluster 'Sani-Cluster' is active. In the 'Tasks' section, there is one pending task with the ID 'fdcc20855c...' and status 'Pending'. The 'Services' section shows a draining service.

10. Now keep on refreshing until the status is running.

The screenshot shows the AWS ECS console with the same URL as the previous screenshot. The task status has now changed to 'Running'. The task details show it was started 48 seconds ago.

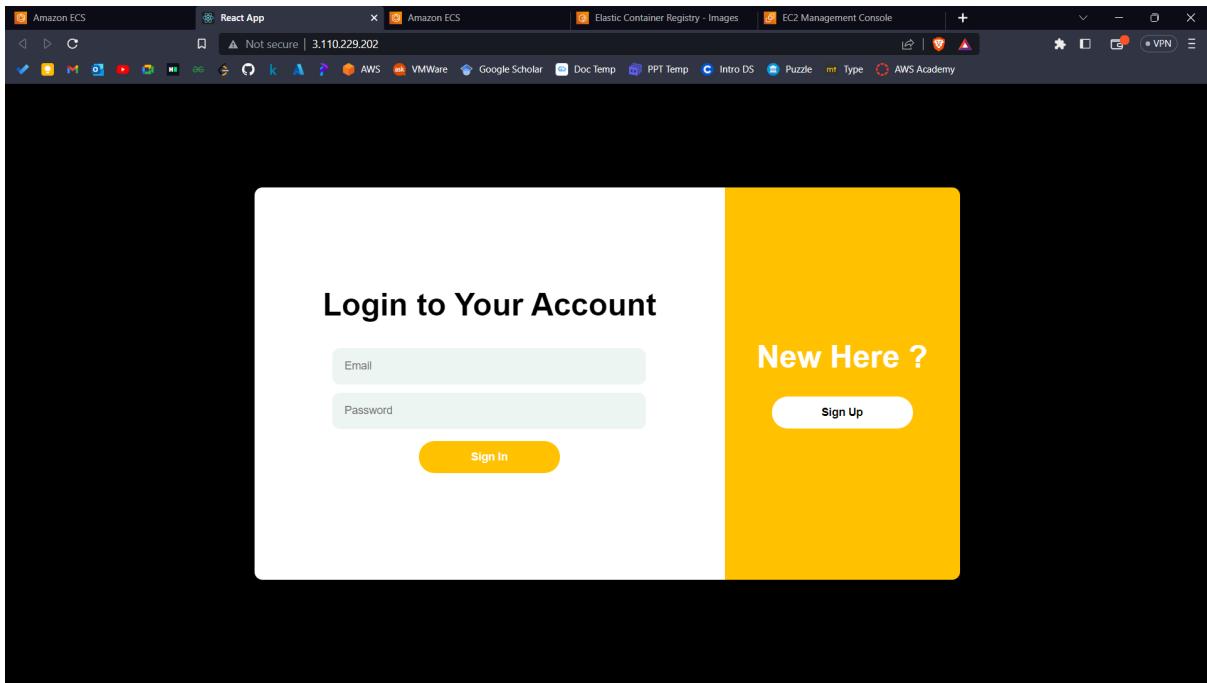
11. Now click on that task.

The screenshot shows the AWS ECS Task Overview page. The task has been successfully launched with the ARN: arn:aws:ecs:ap-south-1:██████████:task/Sani-Cluster/fdec20855cb84fd597d55589eb59953e. The task status is Running. The Configuration tab is selected, showing the container details for Sani-Login. The container's image URI is trial-ecr:login, it is essential, and there is no command specified. The Network bindings tab is also visible, showing a single binding where port 80 on the host maps to port 80 on the container, using the TCP protocol.

12. Go to network bindings.

This screenshot is identical to the previous one, showing the AWS ECS Task Overview page for the same task. The task status is now explicitly shown as Running. The Network bindings tab is selected, providing a detailed view of the port mapping: Host port 80 maps to Container port 80 via TCP. An external link is provided to access the service at 3.110.229.202:80.

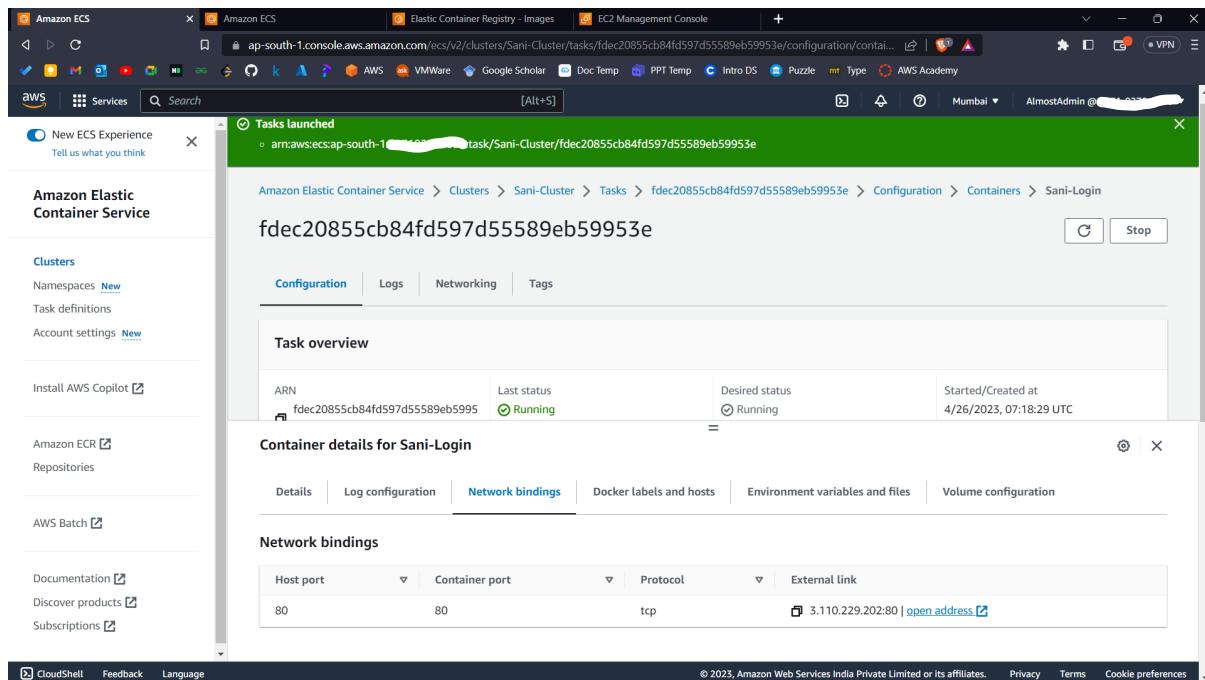
13. Click on open address.



Hurray! Your application is up and running.

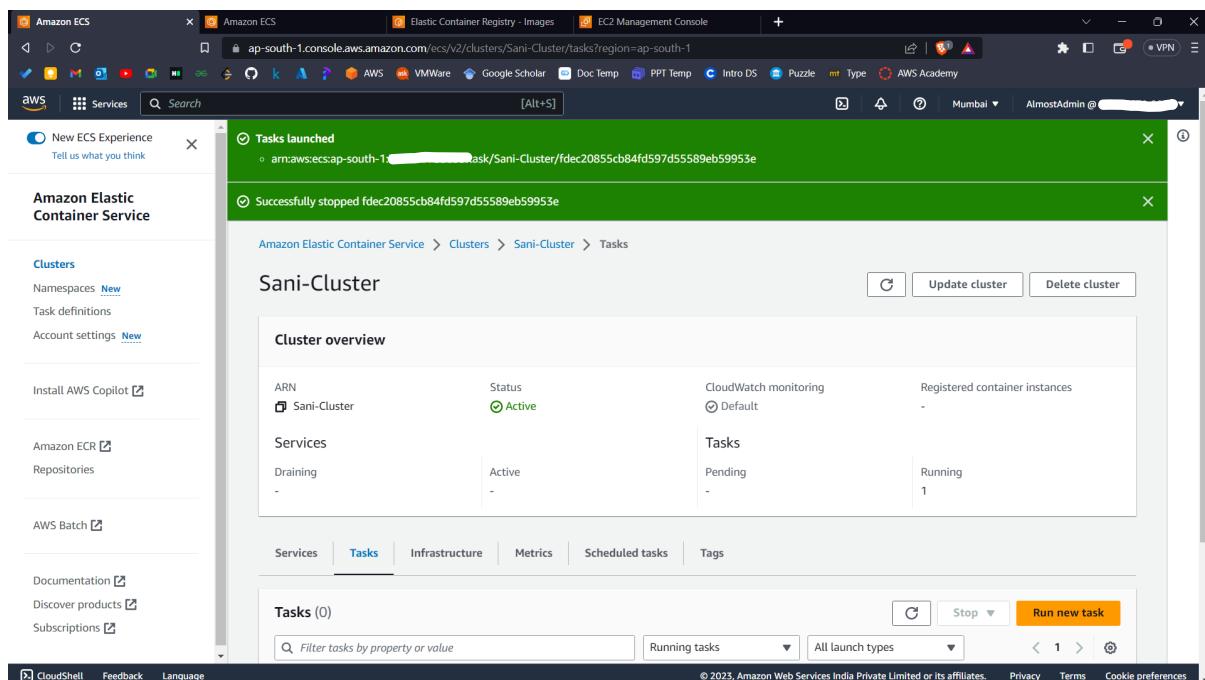
Part 4: - Delete all the services.

1. Click stop to stop this task.



The screenshot shows the AWS Elastic Container Service (ECS) console. On the left, there's a sidebar with options like 'Clusters', 'Namespaces', 'Task definitions', and 'Account settings'. The main area displays a task named 'fdec20855cb84fd597d55589eb59953e'. The task status is 'Running'. Below the task card, there's a 'Container details for Sani-Login' section with tabs for 'Details', 'Log configuration', 'Network bindings', 'Docker labels and hosts', 'Environment variables and files', and 'Volume configuration'. The 'Network bindings' tab is selected, showing a host port of 80 mapped to a container port of 80 over TCP, with an external link to 3.110.229.202:80.

2. Click on delete cluster.



The screenshot shows the AWS ECS console with the 'Clusters' section selected. It displays a cluster named 'Sani-Cluster' which is currently 'Active'. The 'Tasks' tab is selected under the cluster overview. At the bottom of the cluster card, there are buttons for 'Update cluster' and 'Delete cluster'. Below the cluster card, there's a 'Tasks (0)' section with a search bar and filters for 'Running tasks' and 'All launch types'.

The screenshot shows the AWS Elastic Container Service (ECS) console. On the left, there's a sidebar with links like 'Clusters', 'Namespaces', 'Task definitions', and 'Account settings'. The main area shows a green success message: 'Tasks launched' (with one task listed), 'Successfully stopped fdec20855cb84fd597d55589eb59953e', and 'Cluster Sani-Cluster has been deleted successfully.' Below this, the 'Clusters' section shows a table with columns: Cluster, Services, Tasks, Registered container instances, CloudWatch monitoring, and Capacity provider strategy. The table is currently empty, displaying 'No clusters' and 'No clusters to display'.

3. Now go to task definitions: -

This screenshot shows the 'Task definitions' page in the AWS ECS console. The sidebar on the left is identical to the previous screenshot. The main area displays a green success message: 'Task definition successfully created' (with the note 'Sani-Login-Task1 has been successfully created. You can use this task definition to deploy a service or run a task.') and a 'Deploy' button. Below this, the 'Task definitions' table shows one entry: 'Sani-Login-Task' with a status of 'ACTIVE'. There are filters at the top of the table, including 'Status of last revision = ACTIVE' and a 'Clear filters' button.

4. Open your task definition.

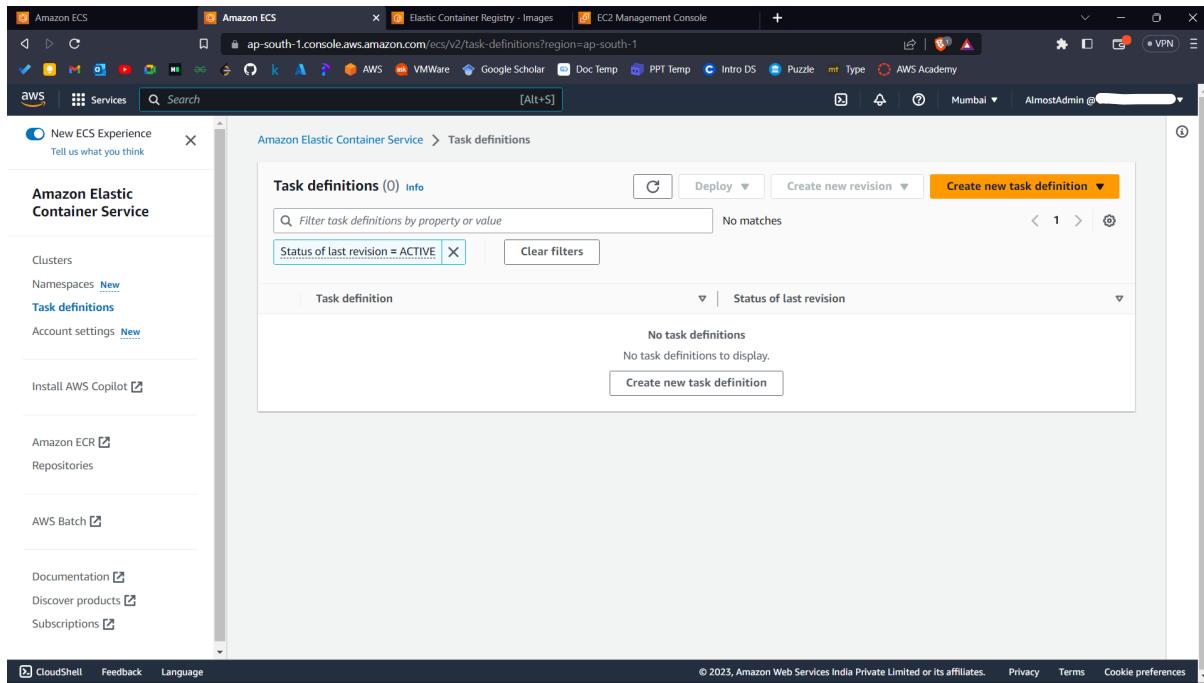
The screenshot shows the AWS Elastic Container Service (ECS) Task Definitions page. A green success message at the top states: "Task definition successfully created" and "Sani-Login-Task:1 has been successfully created. You can use this task definition to deploy a service or run a task." The main table displays one task definition:

Task definition: revision	Status
Sani-Login-Task:1	ACTIVE

5. Select your revision and go to actions and deregister.

The screenshot shows the same AWS ECS Task Definitions page as before, but the Actions dropdown menu is open. The "Deregister" option is highlighted with a blue selection bar. The rest of the interface is identical to the previous screenshot, showing the successful creation of the task definition.

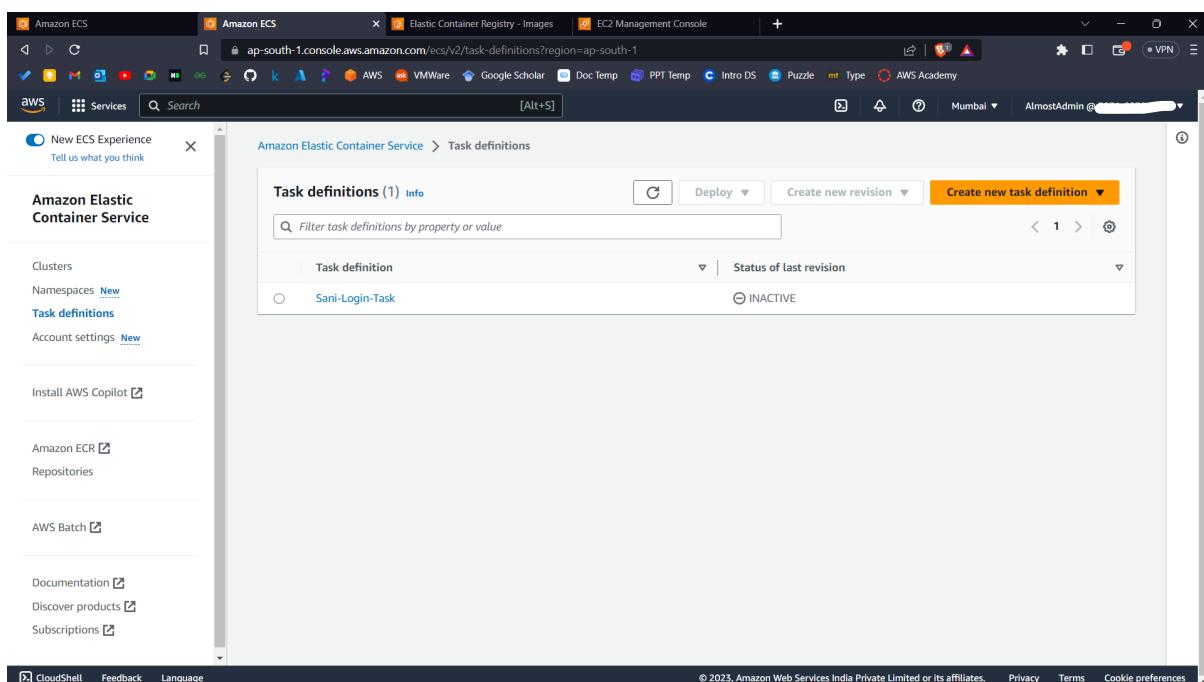
6. Again, go to task definition.



The screenshot shows the AWS Elastic Container Service (ECS) Task Definitions page. The left sidebar is titled "Amazon Elastic Container Service" and includes links for Clusters, Namespaces, Task definitions (which is currently selected), Account settings, and other services like Amazon ECR and AWS Batch. The main content area has a header "Task definitions (0) Info". Below it is a search bar and a filter section with the status set to "ACTIVE". A large message states "No task definitions" and "No task definitions to display." At the bottom right of the main area is a button labeled "Create new task definition".

7. Clear the filter.

Go inside the task definition.



This screenshot shows the same AWS ECS Task Definitions page after clearing the filter. The main content area now displays a single task definition named "Sani-Login-Task" with a status of "INACTIVE". The rest of the interface remains identical to the previous screenshot, with the left sidebar and the top navigation bar visible.

The screenshot shows the AWS Elastic Container Service (ECS) Task Definitions page. The URL is ap-south-1.console.aws.amazon.com/ecs/v2/task-definitions/Sani-Login-Task?status=INACTIVE®ion=ap-south-1. The left sidebar is titled "Amazon Elastic Container Service" and includes links for Clusters, Namespaces, Task definitions (which is selected), Account settings, and various AWS services like ECR, Batch, and Copilot. The main content area is titled "Sani-Login-Task (1) Info". It shows a table with one row: "Task definition: revision" (Sani-Login-Task:1) and "Status" (INACTIVE). There are buttons for Deploy, Actions (with options for Deregister and Delete), Create new revision, and a search bar.

8. Select it and click actions and delete.

This screenshot is identical to the previous one, showing the "Sani-Login-Task" task definition in the AWS ECS Task Definitions page. The difference is in the "Actions" dropdown menu, which is now open and has the "Delete" option highlighted with a blue border. The other options in the dropdown are "Deregister" and "Create new revision". The rest of the interface and data remain the same.

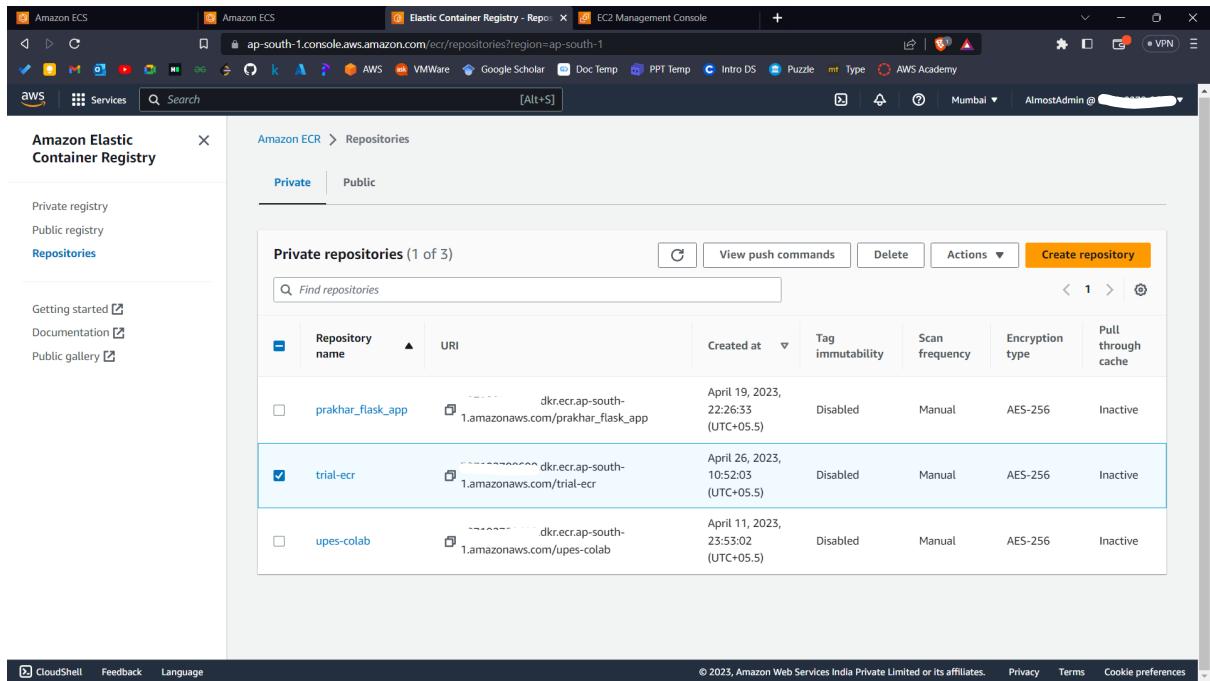
The screenshot shows the AWS Elastic Container Service (ECS) Task Definitions page. A green success message at the top states: "Sani-Login-Task1 has been successfully submitted for deletion." Below this, the task definition "Sani-Login-Task" is listed with a status of "0". A sub-section titled "Task definition: revision" shows the message "No task definitions" and "No task definitions to display." The left sidebar includes links for Clusters, Namespaces, Task definitions (which is selected), Account settings, and various AWS services like ECR and Batch.

9. Now go to ECR.

The screenshot shows the AWS Elastic Container Registry (ECR) Repositories page. It displays three private repositories: "prakhar_flask_app", "trial-ecr", and "upes-colab". Each repository entry includes its name, URI, creation date, tag immutability, scan frequency, encryption type, and pull-through cache status. The left sidebar shows links for Private registry, Public registry, and Repositories (which is selected).

Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type	Pull through cache
prakhar_flask_app	.dkr.ecr.ap-south-1.amazonaws.com/prakhar_flask_app	April 19, 2023, 22:26:33 (UTC+05:5)	Disabled	Manual	AES-256	Inactive
trial-ecr	.dkr.ecr.ap-south-1.amazonaws.com/trial-ecr	April 26, 2023, 10:52:03 (UTC+05:5)	Disabled	Manual	AES-256	Inactive
upes-colab	.dkr.ecr.ap-south-1.amazonaws.com/upes-colab	April 11, 2023, 23:53:02 (UTC+05:5)	Disabled	Manual	AES-256	Inactive

10. Select your repository and delete it.



The screenshot shows the Amazon ECR console with the URL ap-south-1.console.aws.amazon.com/ecr/repositories?region=ap-south-1. The left sidebar shows options like 'Amazon Elastic Container Registry', 'Private registry', 'Public registry', and 'Repositories'. The main area displays 'Private repositories (1 of 3)' with three entries:

Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type	Pull through cache
prakhar_flask_app	dkr.ecr.ap-south-1.amazonaws.com/prakhar_flask_app	April 19, 2023, 22:26:33 (UTC+05:5)	Disabled	Manual	AES-256	Inactive
trial-ecr	dkr.ecr.ap-south-1.amazonaws.com/trial-ecr	April 26, 2023, 10:52:03 (UTC+05:5)	Disabled	Manual	AES-256	Inactive
upes-colab	dkr.ecr.ap-south-1.amazonaws.com/upes-colab	April 11, 2023, 23:53:02 (UTC+05:5)	Disabled	Manual	AES-256	Inactive

All the resources which can result in generating bills have been deleted successfully!

