https://github.com/suvaatnbu/Orchestration-Scaling

Description
Notes:
Jenkins Credentials:

URL: http://3.111.188.91:8080/
Username: herovired
Password: herovired
Project Link: https://github.com/UnpredictablePrashant/SampleMERNwithMicroservices
Fork this repository. For the update from the main repository, you can refer to this link: https://stackoverflow.com/questions/3903817/pull-new-updates-from-original-github-repository-into-forked-github-repository
Project Steps:

Step 1: Set Up the AWS Environment

1. Set Up AWS CLI and Boto3:

   - Install AWS CLI and configure it with AWS credentials.
   - Install Boto3 for Python and configure it.

Step 2: Prepare the MERN Application
1. Containerize the MERN Application:
   - Ensure the MERN application is containerized using Docker. Create a Dockerfile for each component (frontend and backend).
2. Push Docker Images to Amazon ECR:
   - Build Docker images for the frontend and backend.
   - Create an Amazon ECR repository for each image.
   - Push the Docker images to their respective ECR repositories.

Step 3: Version Control
1. Use AWS CodeCommit:
   - Create a CodeCommit repository.
   - Push the MERN application source code to the CodeCommit repository.

Step 4: Continuous Integration with Jenkins
1. Set Up Jenkins:
   - Install Jenkins on an EC2 instance.
   - Configure Jenkins with necessary plugins.
2. Create Jenkins Jobs:
   - Create Jenkins jobs for building and pushing Docker images to ECR.
   - Trigger the Jenkins jobs whenever there's a new commit in the CodeCommit repository.

Step 5: Infrastructure as Code (IaC) with Boto3
1. Define Infrastructure with Boto3 (Python Script):
   - Use Boto3 to define the infrastructure (VPC, subnets, security groups).
   - Define an Auto Scaling Group (ASG) for the backend.
   - Create AWS Lambda functions if needed.

Step 6: Deploying Backend Services

1. Deploy Backend on EC2 with ASG:
   - Use Boto3 to deploy EC2 instances with the Dockerized backend application in the ASG.

Step 7: Set Up Networking

1. Create Load Balancer:
   - Set up an Elastic Load Balancer (ELB) for the backend ASG.
2. Configure DNS:
   - Set up DNS using Route 53 or any other DNS service.

Step 8: Deploying Frontend Services

1. Deploy Frontend on EC2:
   - Use Boto3 to deploy EC2 instances with the Dockerized frontend application.

Step 9: AWS Lambda Deployment

1. Create Lambda Functions:

- Use Boto3 to create AWS Lambda functions for specific tasks within the application.
- Backup of Db using Lambda Functions and store in S3 bucket - put time stamping on the backup

Step 10: Kubernetes (EKS) Deployment

1. Create EKS Cluster:
   - Use eksctl or other tools to create an Amazon EKS cluster.
2. Deploy Application with Helm:
   - Use Helm to package and deploy the MERN application on EKS.

Step 11: Monitoring and Logging

1. Set Up Monitoring:
   - Use CloudWatch for monitoring and setting up alarms.

2. Configure Logging:
   - Use CloudWatch Logs or another logging solution for collecting logs.

Step 12: Documentation

1. Document the Architecture:

- Instruct learners to create documentation for the entire architecture and deployment process.
- Put everything on the GitHub

Step 13: Final Checks

1. Validate the Deployment:
   - Ensure that the MERN application is accessible and functions correctly.

BONUS: ChatOps

Step 14: ChatOps Integration

Create SNS Topics:

1. Use Boto3 to create SNS topics for different events (e.g., deployment success, failure).

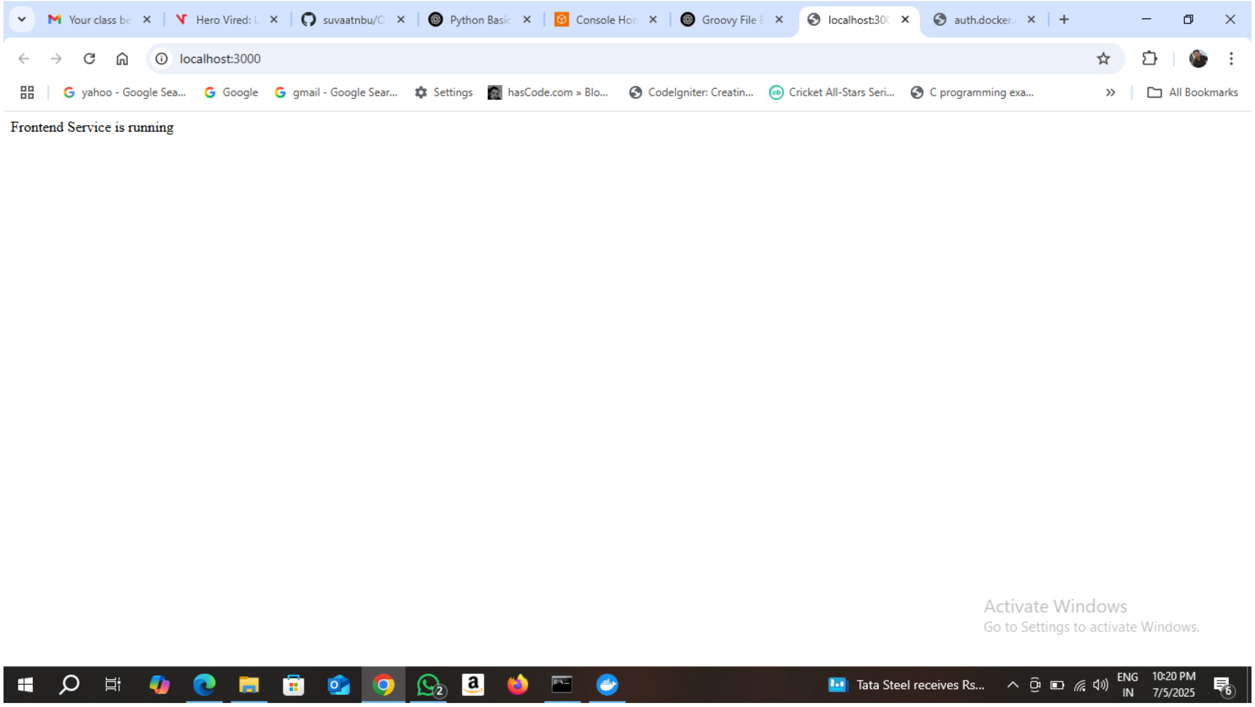
2. Create Lambda for ChatOps:
1. Write a Lambda function that sends notifications to the appropriate SNS topics based on deployment events.
2. Integrate ChatOps with Messaging Platform:
3. Configure integrations with a messaging platform (e.g., Slack/MS Teams/ Telegram) to receive notifications from SNS.
4. Configure SES

Submission Instructions:

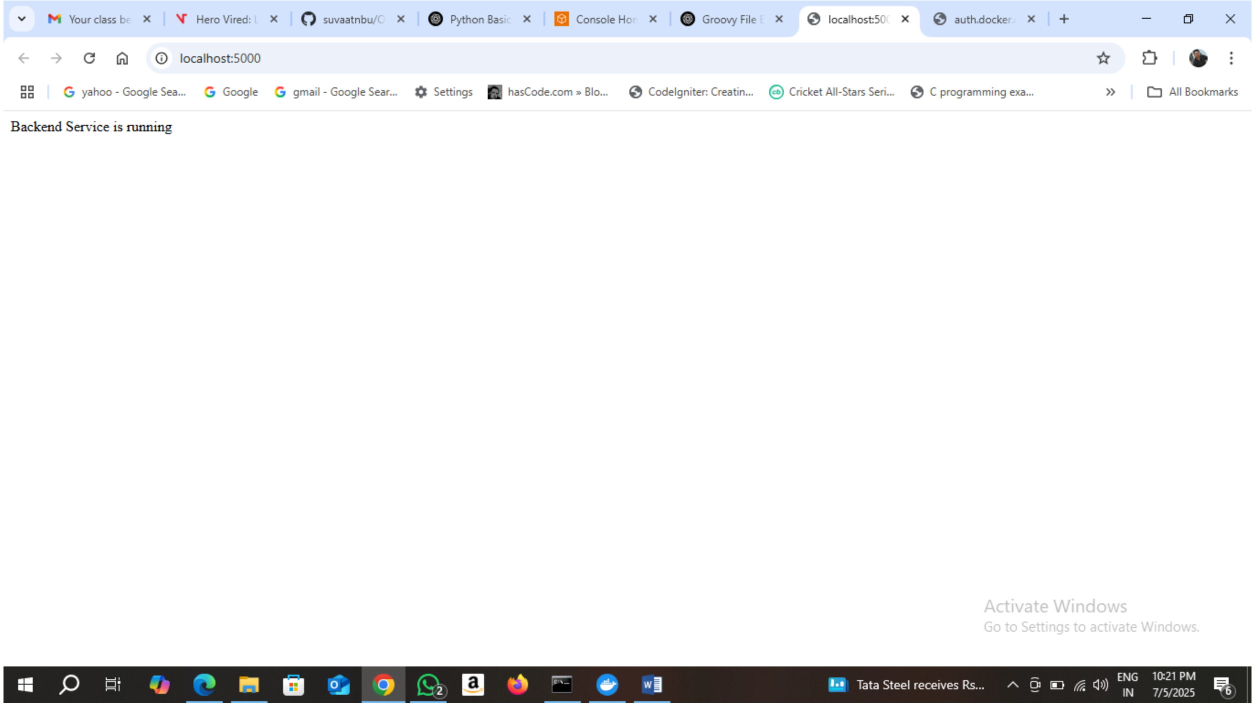
To submit your assignment, please follow these guidelines:

- Ensure that your assignment is fully completed.
- Push your code to a GitHub repository.
- Share the repository link by including it in a text, Word, or PDF file format.

Submit the file/text containing the repository link via Vlearn.

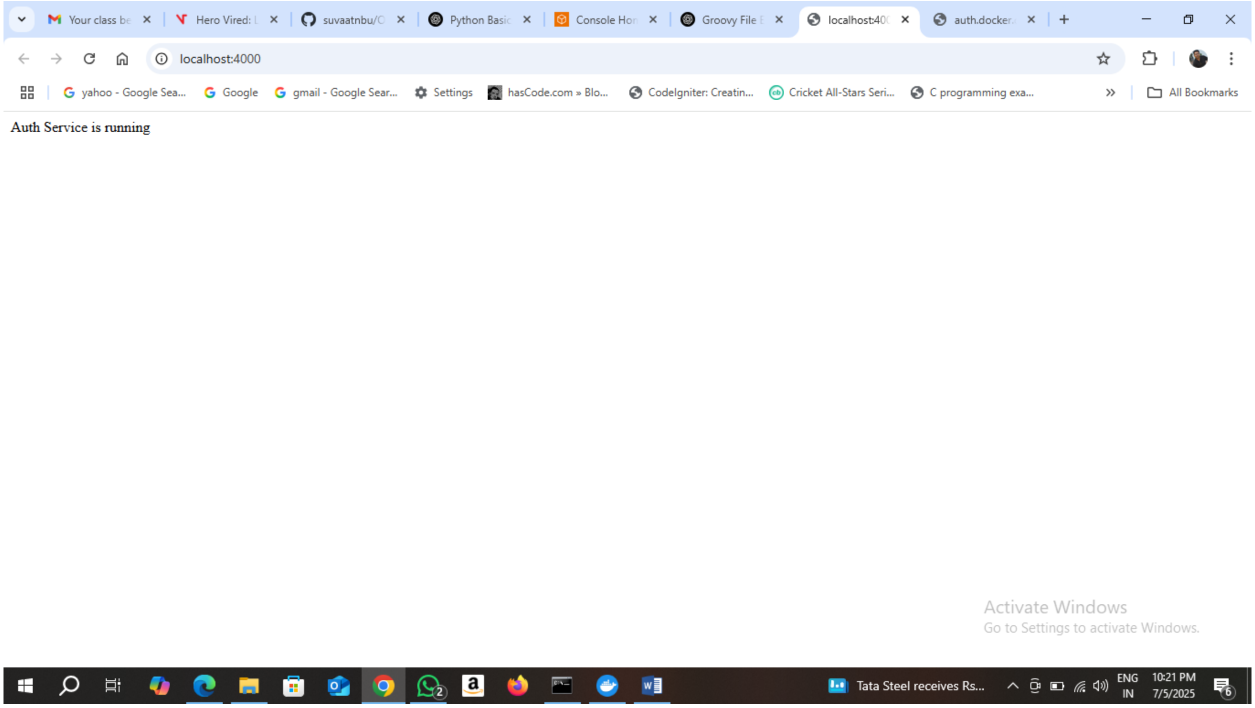


Frontend Service is running



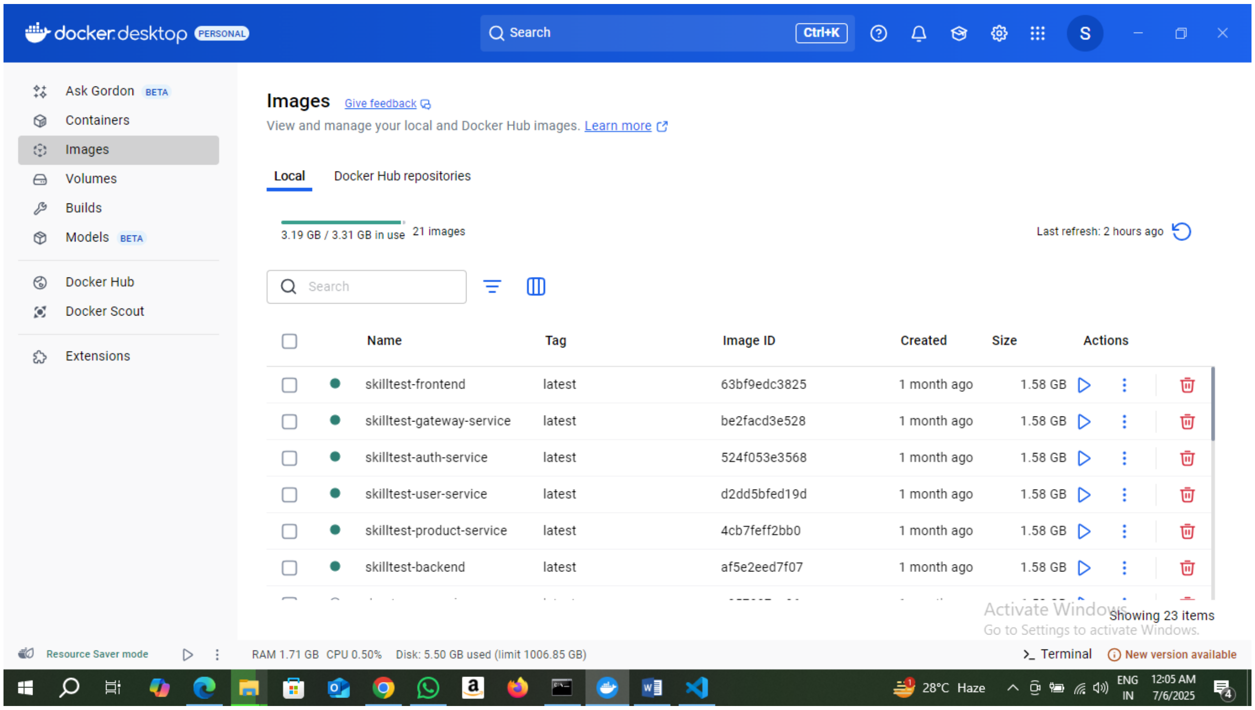
localhost:5000

Backend Service is running



localhost:4000

Auth Service is running



Images

Give feedback

View and manage your local and Docker Hub images. Learn more

Local | Docker Hub repositories

3.19 GB / 3.31 GB in use 21 images

Last refresh: 2 hours ago

Search

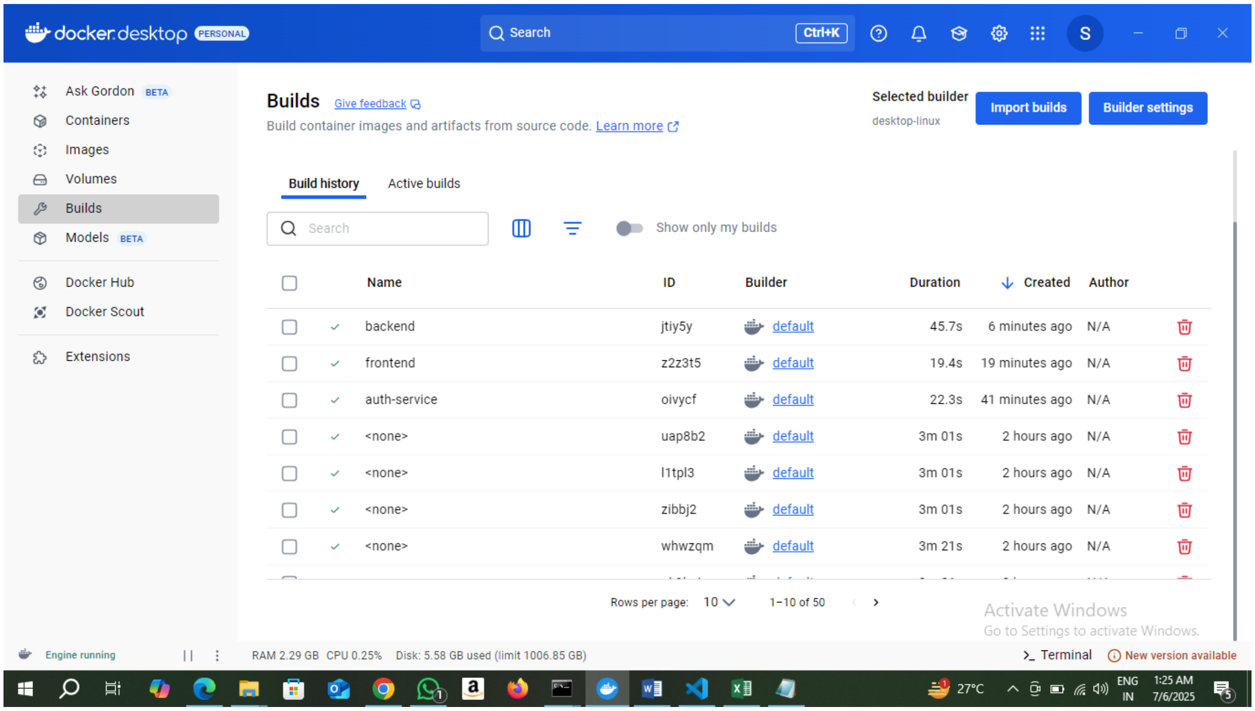
□		Name	Tag	Image ID	Created	Size	Actions
□	●	skilltest-frontend	latest	63bf9edc3825	1 month ago	1.58 GB	
□	●	skilltest-gateway-service	latest	be2facd3e528	1 month ago	1.58 GB	
□	●	skilltest-auth-service	latest	524f053e3568	1 month ago	1.58 GB	
□	●	skilltest-user-service	latest	d2dd5bfed19d	1 month ago	1.58 GB	
□	●	skilltest-product-service	latest	4cb7feff2bb0	1 month ago	1.58 GB	
□	●	skilltest-backend	latest	af5e2eed7f07	1 month ago	1.58 GB	

Showing 23 items

Activate Windows
Go to Settings to activate Windows.

Resource Saver mode RAM 1.71 GB CPU 0.50% Disk: 5.50 GB used (limit 1006.85 GB) Terminal New version available

28°C Haze ENG IN 12:05 AM 7/6/2025



Builds

Give feedback

Build container images and artifacts from source code. Learn more

Selected builder
desktop-linux

Import builds | Builder settings

Build history | Active builds

Search

Show only my builds

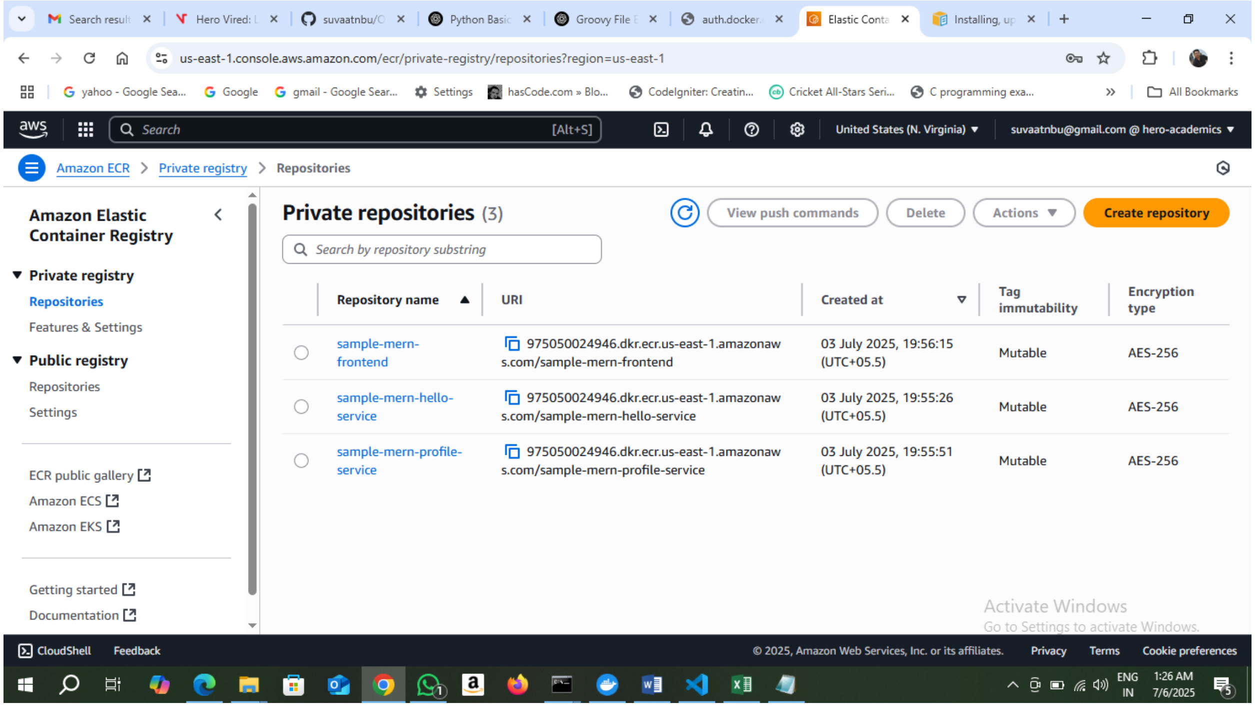
□		Name	ID	Builder	Duration	Created	Author
□	✓	backend	jtiy5y	default	45.7s	6 minutes ago	N/A
□	✓	frontend	z2z3t5	default	19.4s	19 minutes ago	N/A
□	✓	auth-service	oivycf	default	22.3s	41 minutes ago	N/A
□	✓	<none>	uap8b2	default	3m 01s	2 hours ago	N/A
□	✓	<none>	l1tpl3	default	3m 01s	2 hours ago	N/A
□	✓	<none>	zibbj2	default	3m 01s	2 hours ago	N/A
□	✓	<none>	whwzqm	default	3m 21s	2 hours ago	N/A

Rows per page: 10 1–10 of 50

Activate Windows
Go to Settings to activate Windows.

Engine running RAM 2.29 GB CPU 0.25% Disk: 5.58 GB used (limit 1006.85 GB) Terminal New version available

27°C ENG IN 1:25 AM 7/6/2025



us-east-1.console.aws.amazon.com/ecr/private-registry/repositories?region=us-east-1

Amazon ECR > Private registry > Repositories

Amazon Elastic Container Registry

- Private registry
  - Repositories
  - Features & Settings
- Public registry
  - Repositories
  - Settings

ECR public gallery
Amazon ECS
Amazon EKS
Getting started
Documentation

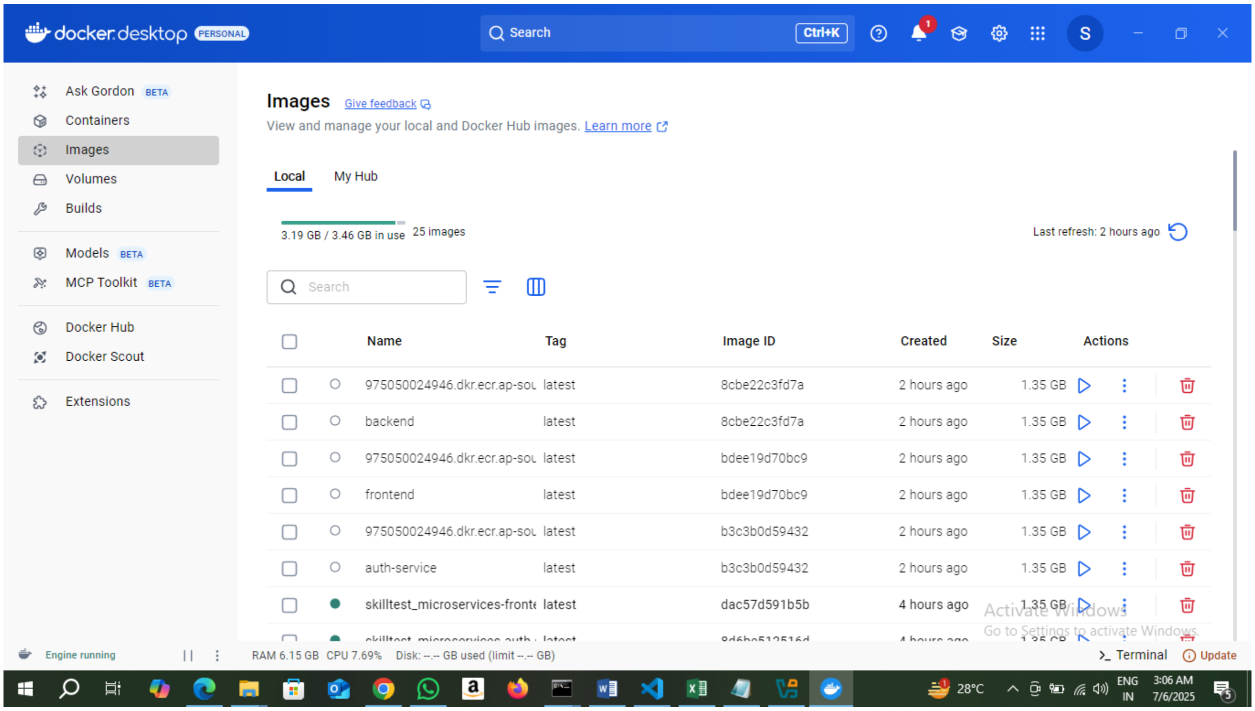
Private repositories (3)

View push commands | Delete | Actions | Create repository

Search by repository substring

Repository name	URI	Created at	Tag immutability	Encryption type
○ sample-mern-frontend	975050024946.dkr.ecr.us-east-1.amazonaws.com/sample-mern-frontend	03 July 2025, 19:56:15 (UTC+05.5)	Mutable	AES-256
○ sample-mern-hello-service	975050024946.dkr.ecr.us-east-1.amazonaws.com/sample-mern-hello-service	03 July 2025, 19:55:26 (UTC+05.5)	Mutable	AES-256
○ sample-mern-profile-service	975050024946.dkr.ecr.us-east-1.amazonaws.com/sample-mern-profile-service	03 July 2025, 19:55:51 (UTC+05.5)	Mutable	AES-256

United States (N. Virginia)
suvaatnbu@gmail.com @ hero-academics
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Activate Windows
Go to Settings to activate Windows.
1:26 AM
7/6/2025



docker desktop PERSONAL

Ask Gordon BETA
Containers
Images
Volumes
Builds
Models BETA
MCP Toolkit BETA
Docker Hub
Docker Scout
Extensions

Images

Give feedback

View and manage your local and Docker Hub images. Learn more

Local | My Hub

3.19 GB / 3.46 GB in use 25 images

Last refresh: 2 hours ago

☐		Name	Tag	Image ID	Created	Size	Actions
☐	○	975050024946.dkr.ecr.ap-sou	latest	8cbe22c3fd7a	2 hours ago	1.35 GB	
☐	○	backend	latest	8cbe22c3fd7a	2 hours ago	1.35 GB	
☐	○	975050024946.dkr.ecr.ap-sou	latest	bdee19d70bc9	2 hours ago	1.35 GB	
☐	○	frontend	latest	bdee19d70bc9	2 hours ago	1.35 GB	
☐	○	975050024946.dkr.ecr.ap-sou	latest	b3c3b0d59432	2 hours ago	1.35 GB	
☐	○	auth-service	latest	b3c3b0d59432	2 hours ago	1.35 GB	
☐	●	skilltest_microservices-fronte	latest	dac57d591b5b	4 hours ago	1.35 GB	
☐	●	skilltest_microservices-auth-	latest	8d6be512516d	4 hours ago	1.35 GB	

Engine running

RAM 6.15 GB CPU 7.69% Disk: --.-- GB used (limit --.-- GB)