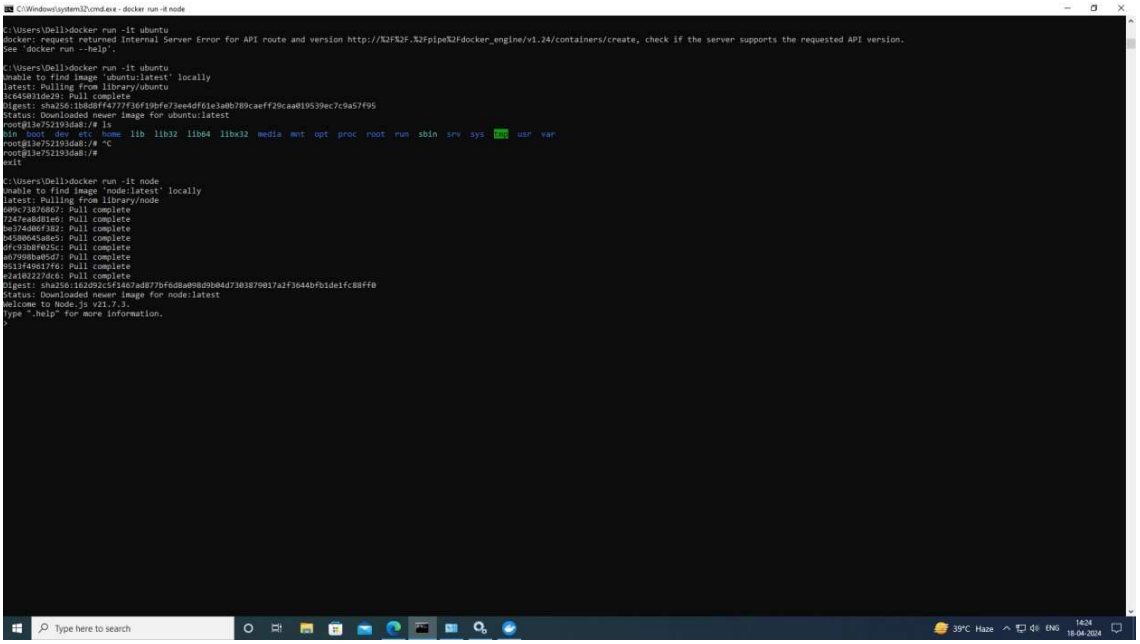
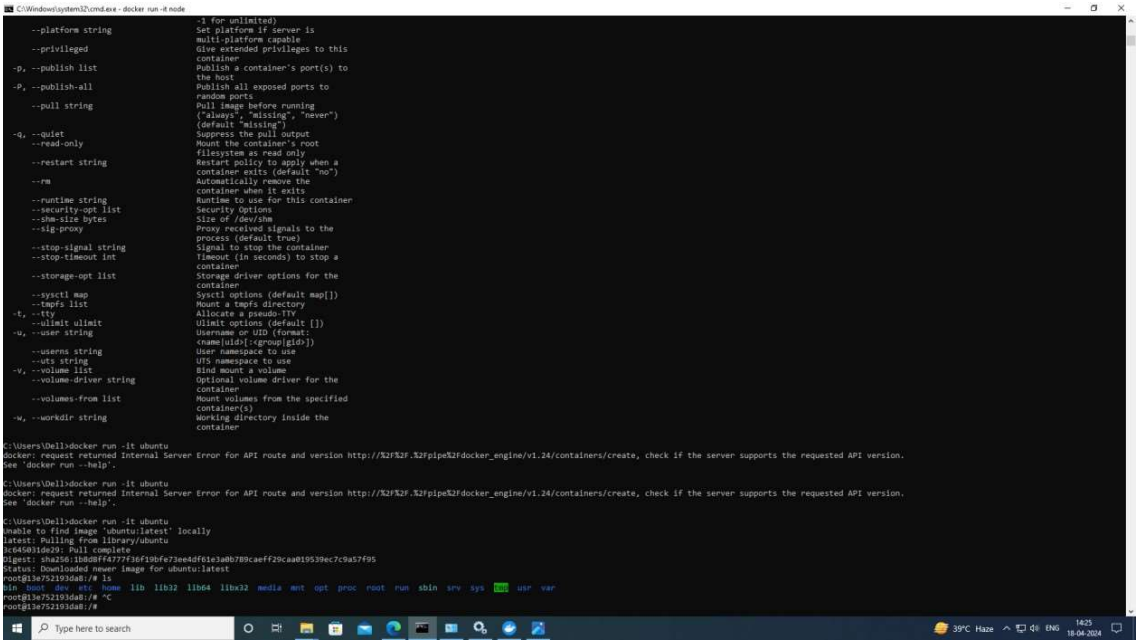
Name: Janhavi Rajesh Bhambare

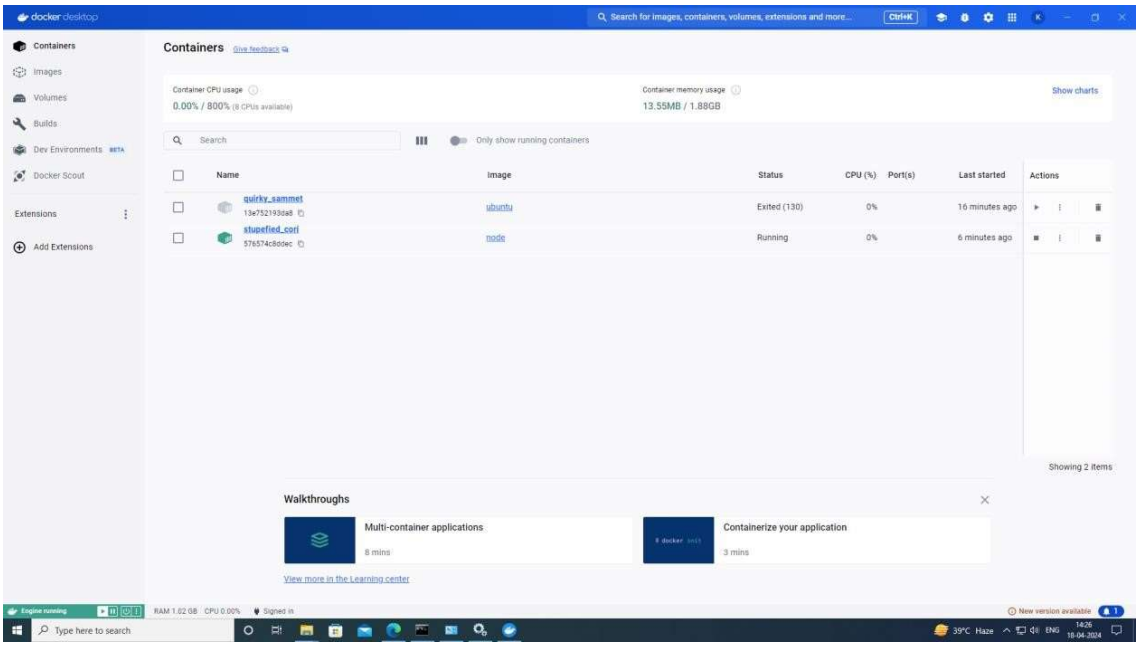
Roll no.:07

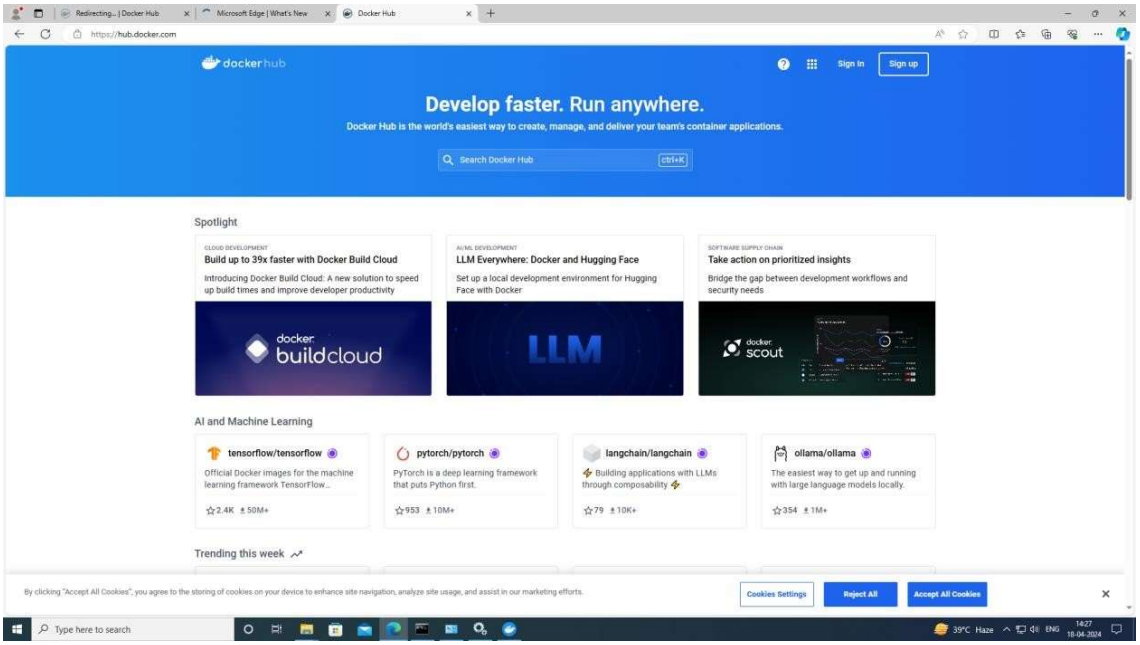
Assignment Name: Docker

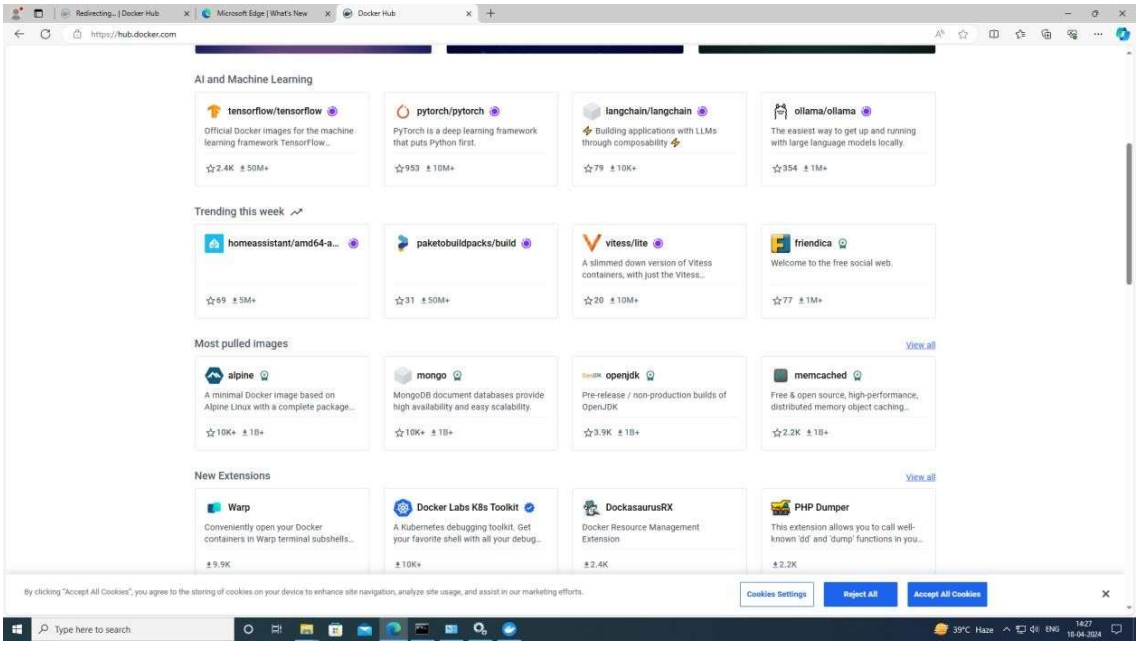
Screenshots:






docker desktop
Search for images, containers, volumes, extensions and more...
Ctrl+K
Containers
Images
Volumes
Builds
Dev Environments BETA
Docker Scout
Extensions
Add Extensions
Containers
Give feedback
Container CPU usage
0.00% / 800% (8 CPUs available)
Container memory usage
13.55MB / 1.88GB
Show charts
Search
Only show running containers
Name
Image
Status
CPU (%)
Port(s)
Last started
Actions
quirky_sammet
13e752193da8
ubuntu
Exited (130)
0%
16 minutes ago
stupefied_cori
576574c8dcec
node
Running
0%
6 minutes ago
Showing 2 items
Walkthroughs
Multi-container applications
8 mins
Containerize your application
3 mins
View more in the Learning center
Engine running
RAM 1.02 GB
CPU 0.00%
Signed in
New version available
Type here to search
39°C Haze
ENG
14:26
18-04-2024


Redirecting... | Docker Hub
Microsoft Edge | What's New
Docker Hub
https://hub.docker.com
dockerhub
Sign in
Sign up
Develop faster. Run anywhere.
Docker Hub is the world's easiest way to create, manage, and deliver your team's container applications.
Search Docker Hub
ctrl+K
Spotlight
CLOUD DEVELOPMENT
Build up to 39x faster with Docker Build Cloud
Introducing Docker Build Cloud: A new solution to speed up build times and improve developer productivity
docker buildcloud
AI/ML DEVELOPMENT
LLM Everywhere: Docker and Hugging Face
Set up a local development environment for Hugging Face with Docker
LLM
SOFTWARE SUPPLY CHAIN
Take action on prioritized insights
Bridge the gap between development workflows and security needs
docker scout
AI and Machine Learning
tensorflow/tensorflow
Official Docker images for the machine learning framework TensorFlow...
☆2.4K ± 50M+
pytorch/pytorch
PyTorch is a deep learning framework that puts Python first.
☆953 ± 10M+
langchain/langchain
Building applications with LLMs through composability
☆79 ± 10K+
ollama/ollama
The easiest way to get up and running with large language models locally.
☆354 ± 1M+
Trending this week
By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.
Cookies Settings
Reject All
Accept All Cookies
Type here to search
39°C Haze
ENG
14:27
18-04-2024



Theory:

Definition and Purpose:

Docker is a platform that allows developers to package applications and their dependencies into lightweight, portable containers.

Containers are isolated environments that encapsulate everything needed to run an application, including code, runtime, system tools, libraries, and settings.

The primary goal of Docker is to streamline the development and deployment of applications by providing a consistent environment across different infrastructure setups.

Key Concepts:

Images: Docker images are read-only templates that contain application code, dependencies, and runtime environment. They are the building blocks used to create containers.

Containers: Containers are instances of Docker images that are running as isolated processes. Each container has its own filesystem, network, and process space, ensuring isolation and portability.

Dockerfile: A Dockerfile is a text file that contains instructions for building a Docker image. It specifies the base image, dependencies, environment variables, and commands needed to set up the application.

Docker Engine: The Docker Engine is the core component of Docker that manages containers, images, networks, and volumes. It includes the Docker daemon (background process) and the Docker CLI (command-line interface).

Docker Registry: Docker registries are repositories for storing and sharing Docker images. The Docker Hub is a public registry hosted by Docker, but private registries can also be used for internal deployments.

Advantages of Docker:

Portability: Docker containers can run on any system with Docker installed, making applications easily portable across different environments (development, testing, production).

Isolation: Containers provide process-level isolation, preventing conflicts between applications and ensuring consistent behavior.

Resource Efficiency: Containers share the host system's kernel, reducing overhead and resource consumption compared to virtual machines.

Scalability: Docker enables easy scaling of applications by creating multiple instances of containers, allowing for efficient resource utilization.

DevOps Integration: Docker facilitates DevOps practices by enabling automated builds, continuous integration/continuous deployment (CI/CD), and infrastructure as code (IaC).