

Name: Janhavi Rajesh Bhambare

Roll no.:07

Assignment Name: Docker

Screenshots:

```
C:\Windows\system32\cmd.exe - docker run -it ubuntu
docker: request returned Internal Server Error for API route and version http://32f32f.32fpipe32fdocker_engine/v1.24/containers/create, check if the server supports the requested API version.
See 'docker run --help'.

C:\Users\Uell\docker run -it ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
1b64501de29: Pull complete
Digest: sha256:1b64501de29f477f31b9bf73e4d6f61ca8b789caeff29caad19539ec7c9a57f95
Status: Downloaded newer image for ubuntu:latest
root@1b64501de29:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run/sbin srv sys usr var
root@1b64501de29:/# ^C
root@1b64501de29:/# exit

C:\Users\Uell\docker run -it node
Unable to find image 'node:latest' locally
latest: Pulling from library/node
696c1876607: Pull complete
7247ea8881ed: Pull complete
6374a606382: Pull complete
64500645a8e3: Pull complete
1f4298079524: Pull complete
4b7908a8e5d7: Pull complete
8631f488176d: Pull complete
1f48e22276d4: Pull complete
Digest: sha256:162892c5f467ad77b6d8a08d9b04d7303879017a2f3044b7b1de1c88ff0
Status: Downloaded newer image for node:latest
Welcome to Node.js v12.7.3.
Type ".help" for more information.
```

```
C:\Windows\system32\cmd.exe - docker run -it node
--platform string          --[1] for unlimited
                          Set platform if server is
                          multi-platform capable
--privileged               Give extended privileges to this
                          container
--publish list             Publish a container's port(s) to
                          the host
--publish-all             Publish all exposed ports to
                          random ports
--pull string              Pull image before running
                          ("always", "missing", "never")
                          (default "missing")
-q, --quiet                Suppress the pull output
--read-only                Mount the container's root
                          filesystem as read only
--restart string           Restart policy to apply when a
                          container exits (default "no")
--rm                       Automatically remove the
                          container when it exits
--runtime string           Runtime to use for this container
--security-opt list        Security Options
--shm-size bytes           Size of /dev/shm
--sig-proxy                Proxy received signals to the
                          process (default true)
--stop-signal string       Signal to stop the container
--stop-timeout int         Timeout (in seconds) to stop a
                          container
--storage-opt list         Storage driver options for the
                          container
--sysctl map               Sysctl options (default map[])
--tmpfs list               Mount a tmpfs directory
-t, --tty                 Allocate a pseudo-TTY
--ulimit ulimit            Ulimit options (default [])
-u, --user string          Username or UID (format:
                          <name>[:uid[:<group>[:gid]])
--users string             User namespace to use
--uts string               UTS namespace to use
-v, --volume list          Bind mount a volume
--volume-driver string     Optional volume driver for the
                          container
--volumes-from list        Mount volumes from the specified
                          container(s)
--workdir string           Working directory inside the
                          container

C:\Users\Uell\docker run -it ubuntu
docker: request returned Internal Server Error for API route and version http://32f32f.32fpipe32fdocker_engine/v1.24/containers/create, check if the server supports the requested API version.
See 'docker run --help'.

C:\Users\Uell\docker run -it ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
1b64501de29: Pull complete
Digest: sha256:1b64501de29f477f31b9bf73e4d6f61ca8b789caeff29caad19539ec7c9a57f95
Status: Downloaded newer image for ubuntu:latest
root@1b64501de29:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run/sbin srv sys usr var
root@1b64501de29:/# ^C
root@1b64501de29:/#
```

Containers

Images

Volumes

Builds

Dev Environments

Docker Scout

Extensions

Add Extensions

Containers

Container CPU usage

Container memory usage

Search

Only show running containers

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
quirky_sammet	ubuntu	Exited (130)	0%		10 minutes ago	
skupified_sori	node	Running	0%		6 minutes ago	

Showing 2 items

Walkthroughs

Multi-container applications

Containerize your application

View more in the Learning center

Engine running

RAM 1.02 GB

CPU 0.00%

Signed in

39°C

Haze

10:27

10-04-2024

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

CLINK

Spotlight

Build up to 35x faster with Docker Build Cloud

Introducing Docker Build Cloud: A new solution to speed up build times and improve developer productivity

docker buildcloud

LLM Everywhere: Docker and Hugging Face

Set up a local development environment for Hugging Face with Docker

LLM

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

5.50M+

pytorch/pytorch

PyTorch is a deep learning framework that puts Python first.

953

8.10M+

langchain/langchain

Building applications with LLMs through composability

79

8.10K+

ollama/ollama

The easiest way to get up and running with large language models locally.

354

8.10M+

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

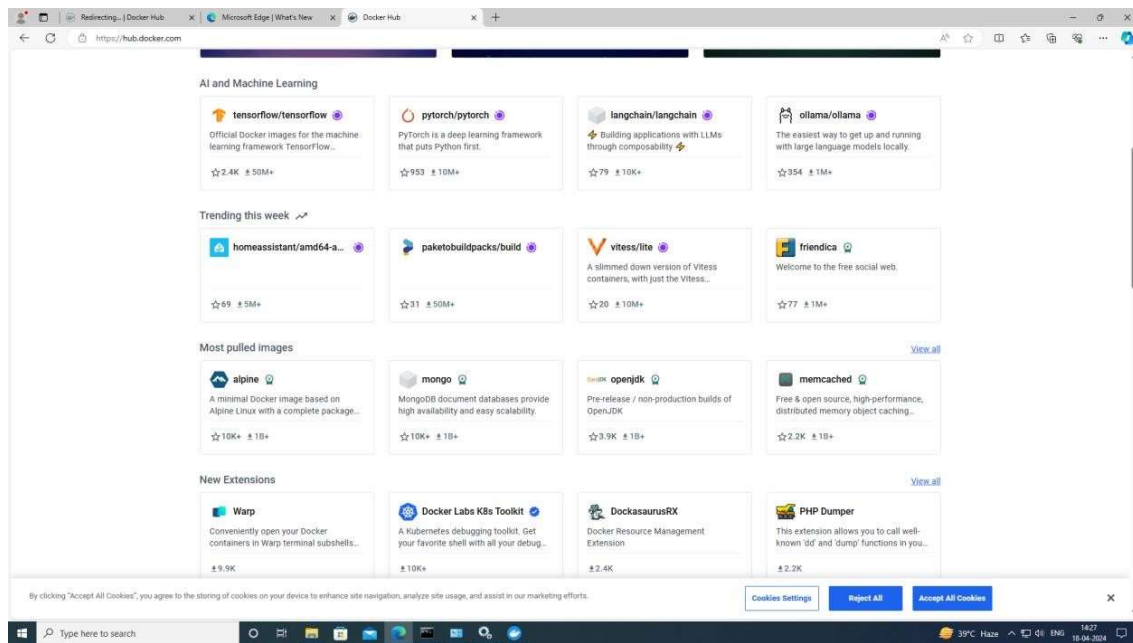
Type here to search

39°C

Haze

10:27

10-04-2024



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).