Sanjeev Agrawal Global Educational University, Bhopal

Lab File of

Containerization using Dockers

Course Code: CA21B403

Prepared by:

Aditya Nair

School of Computer Application

Session: Spring 2024-25

Index

Sr. No.	Title of Experiment	Scheduled date	Actual date	Remarks
1	Installation of Docker on your base OS using Docker toolbox.			
2	Create an account with docker hub and login.			
3	Install and Launch Docker Desktop (use Docker Desktop Installer) over guest OS (ex-Windows 10) running on a VM with the help of Vmware Workstation.			
4	After logging into Docker Hub, explore and customize settings of docker desktop in your virtual machine to facilitate communication between client(CLI) and server(Docker daemon) by enabling experimental features			
5	Using command line, test and validate the functioning of docker using some basic commands: (a) docker version (b) docker help (c) docker info			
6	As a client, pull and run default docker image 'hello-world' on CLI. Once done, check the following: a. status of container formed at runtime using 'docker info' command b. Get details of image id, size and when image was created.			
7	Log into Docker Hub account with your docker id and create a public repository			
8	Create a Custom Image from a Dockerfile.			
9	Explore kubernetes: a) GKE on google cloud platform b) EKS on AWS			
10	Enable kubernetes in docker desktop and check the working of kubernetes by validating with commands from https://docs.docker.com/get-started/orchestration/ .			
11	Log into play with kubernetes : https://labs.play-with-k8s.com/ with either docker hub or git hub account. Try the steps of kubernetes handson workshop : https://training.play-with-kubernetes.com/kubernetes-workshop/			

EXPERIMENT -1

Title: Installation of Docker on your base OS using Docker toolbox. Theory:

Following steps are needs to follow:

1. Check System Requirements:

Make sure your system meets the minimum requirements for running Docker Toolbox. You need a 64-bit version of Windows 10, at least 4 GB of RAM, and Virtualization Technology enabled in BIOS.

2. Download Docker Toolbox: Go to the Docker Toolbox GitHub releases page (https://github.com/docker/toolbox/releases) and download the latest version of Docker Toolbox for Windows.

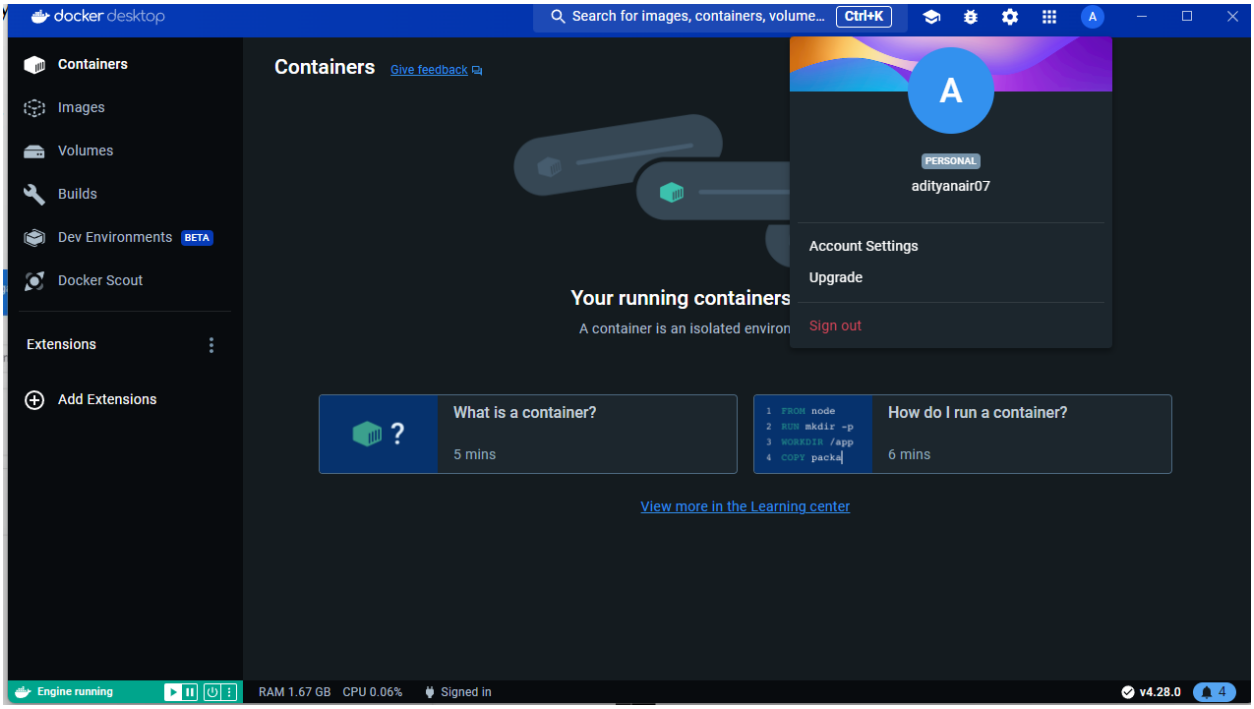
3. Install Docker Toolbox:

Double-click the downloaded installer to start the installation process. Follow the on- screen instructions to complete the installation.

4. Launch Docker Toolbox: After the installation is complete, launch Docker Toolbox from the Start menu. It will set up a Docker environment for you, including creating a VirtualBox VM called "default".

5. Initialize Docker Environment: Once Docker Toolbox is launched, it will open a terminal window. Wait for it to finish setting up the environment. It will output some information once it's done.

6. Test Docker Installation: Open a command prompt or PowerShell window and type docker version to verify that Docker is installed correctly. You should see the version information for both the client and the server


docker desktop
Search for images, containers, volume...
Ctrl+K
Containers
Images
Volumes
Builds
Dev Environments
BETA
Docker Scout
Extensions
Add Extensions
Containers
Give feedback
PERSONAL
adityanair07
Account Settings
Upgrade
Sign out
Your running containers
A container is an isolated environ
What is a container?
5 mins
How do I run a container?
6 mins
View more in the Learning center
Engine running
RAM 1.67 GB CPU 0.06%
Signed in
v4.28.0

Experiment -2

Title: Create an account with docker hub and login.

1. Go to Docker Hub: Open your web browser and go to the Docker Hub website: hub.docker.com.

2. Sign Up: On the Docker Hub homepage, you should see a "Sign Up" button at the top right corner. Click on it to begin the registration process.

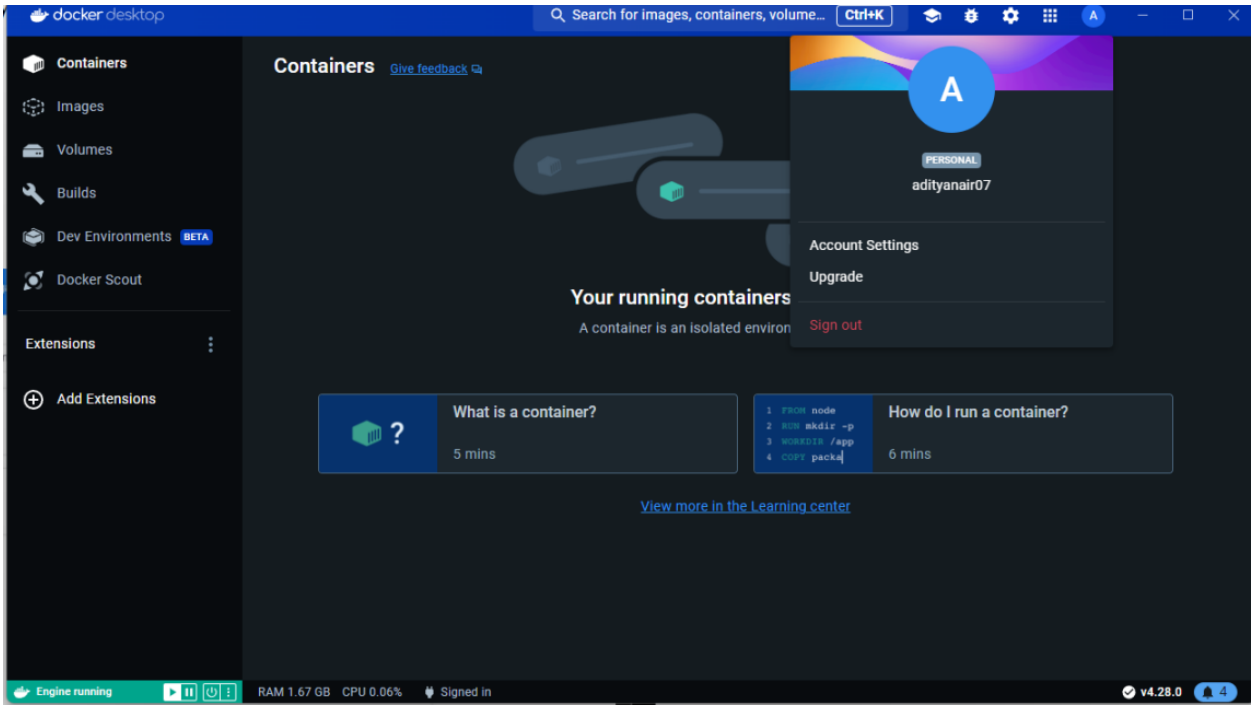
3. Fill in Details: You'll be asked to provide details such as your username, email address, and password. Fill in the required fields and click on "Sign Up for Docker Hub".

4. Verification: After submitting the sign-up form, you might need to verify your email address. Check your email inbox for a verification email from Docker Hub and follow the instructions provided to verify your account

5. Login: Once your account is verified, go back to the Docker Hub website. Click on the "Log In" button at the top right corner.

6. Enter Credentials: Enter your username and password that you used during the sign-up process.

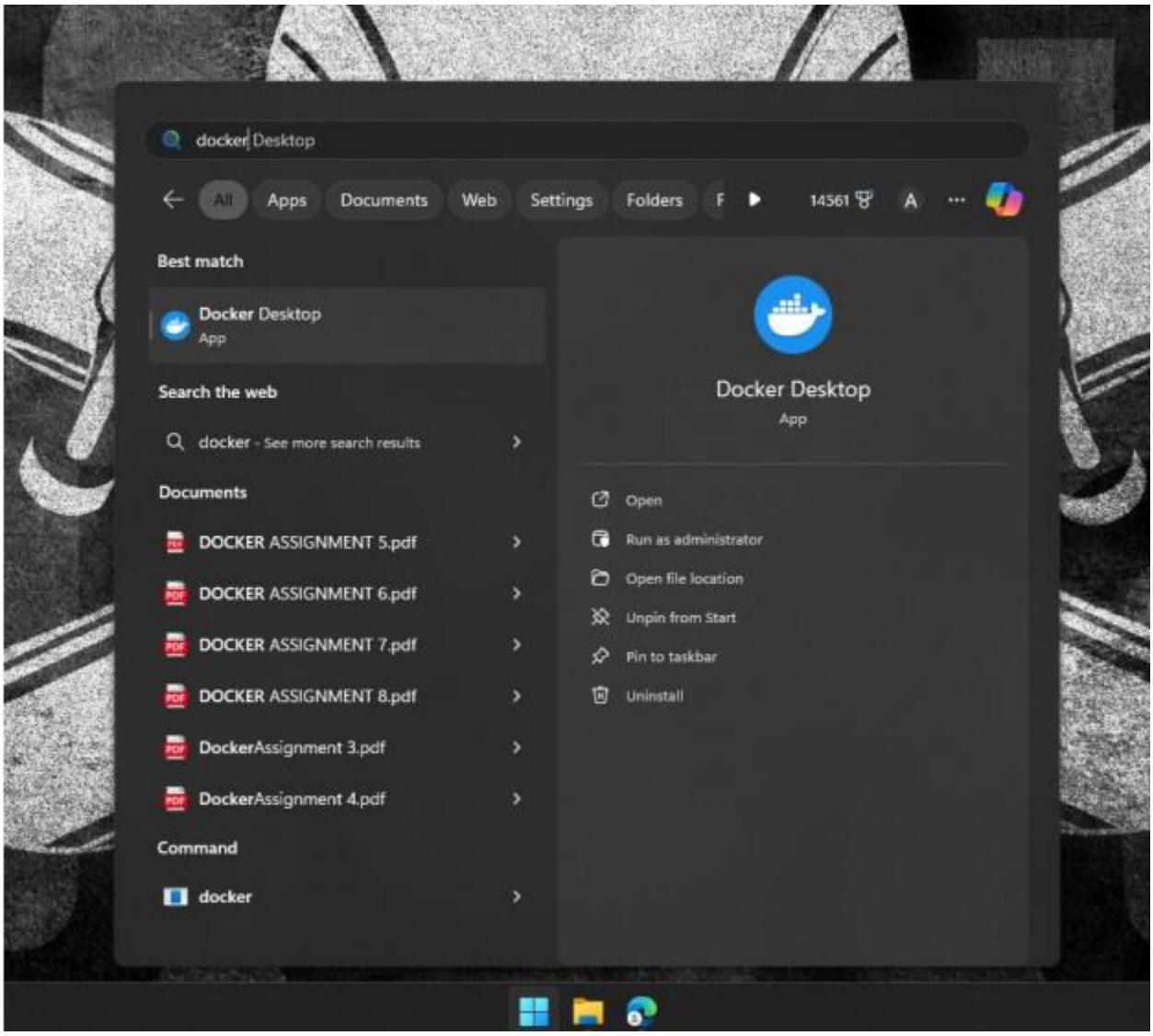
7. Login: Click on the "Log In" button, and you should now be logged in to your Docker Hub account

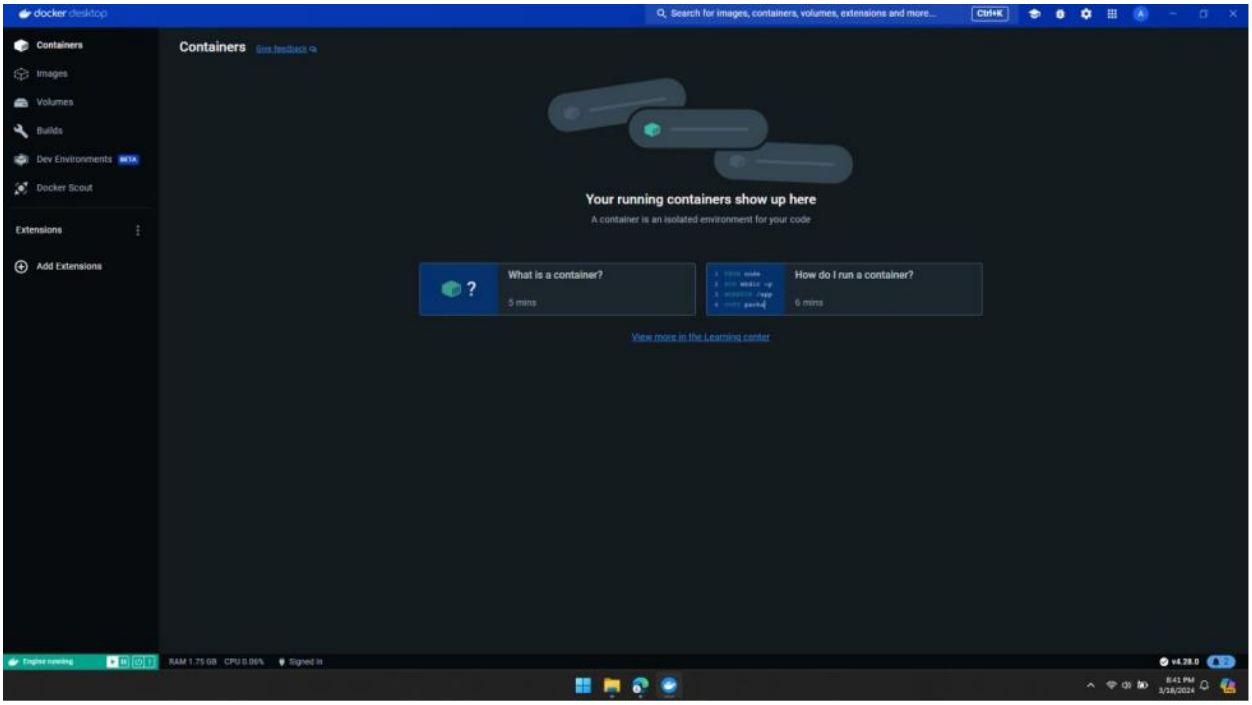

docker desktop
Search for images, containers, volume...
Ctrl+K
Containers
Images
Volumes
Builds
Dev Environments
BETA
Docker Scout
Extensions
Add Extensions
Containers
Give feedback
A
PERSONAL
adityanair07
Account Settings
Upgrade
Sign out
Your running containers
A container is an isolated environ
What is a container?
5 mins
How do I run a container?
6 mins
View more in the Learning center
Engine running
RAM 1.67 GB CPU 0.06%
Signed in
v4.28.0
4

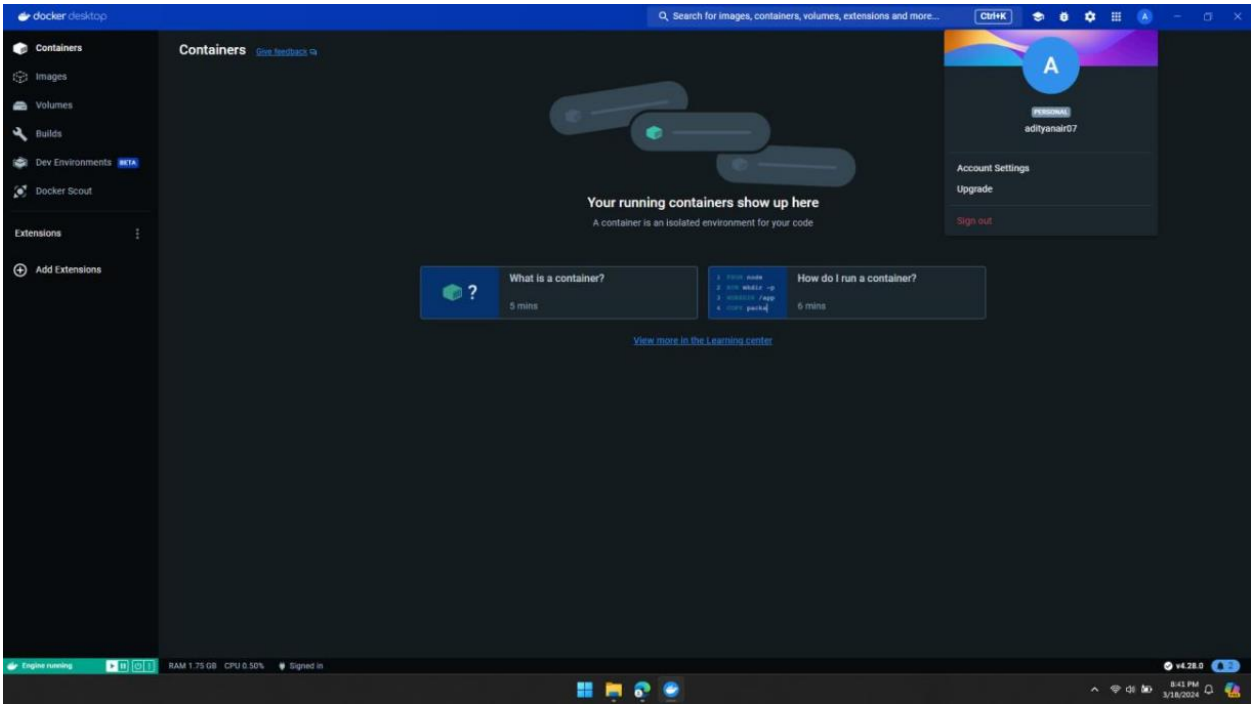
Experiment -3

1. Go to https://www.docker.com/products/docker-desktop/

2. Click to download docker desktop with respect to your host OS



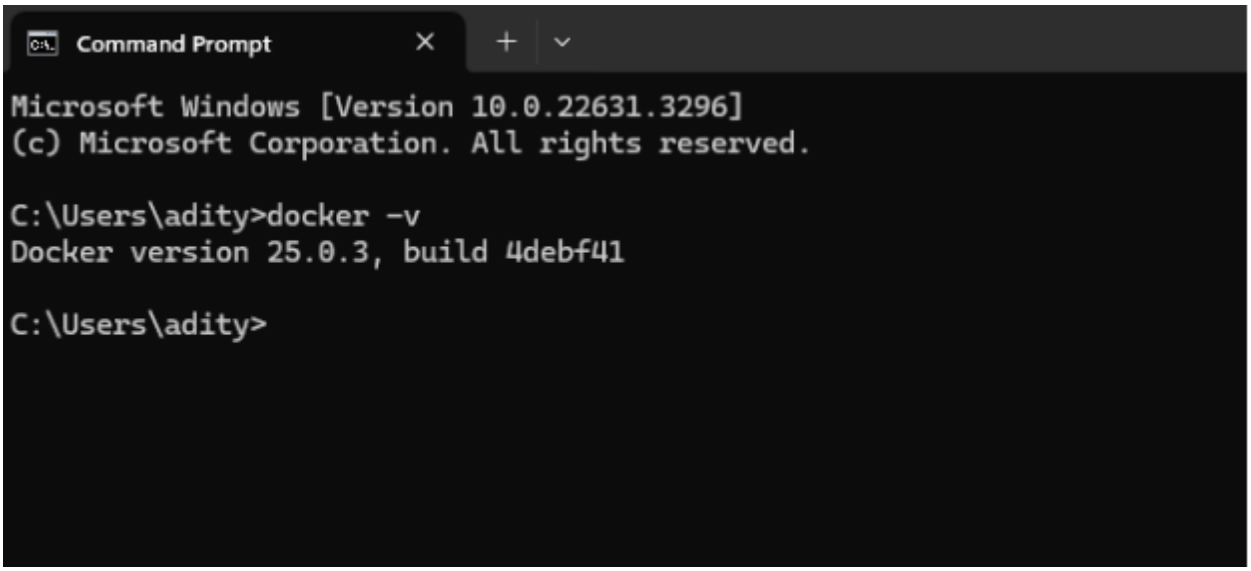

docker desktop
Search for images, containers, volumes, extensions and more...
Ctrl+K
Containers
Images
Volumes
Builds
Dev Environments BETA
Docker Scout
Extensions
Add Extensions
Containers
Give feedback
Your running containers show up here
A container is an isolated environment for your code
What is a container?
5 mins
How do I run a container?
6 mins
View more in the Learning center
Engine running
RAM 1.75 GB CPU 0.06%
Signed in
v4.28.0

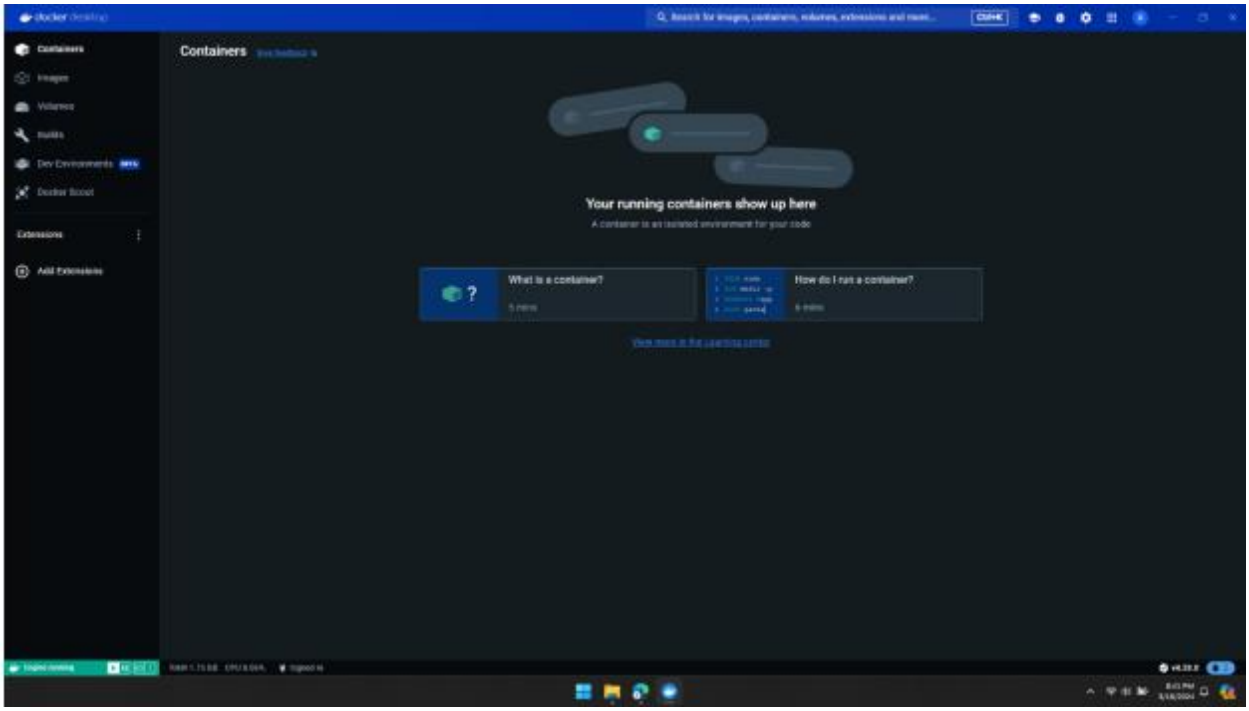

docker desktop
Search for images, containers, volumes, extensions and more...
Ctrl+K
Containers
Images
Volumes
Builds
Dev Environments BETA
Docker Scout
Extensions
Add Extensions
Containers
Give feedback
A
PERSONAL
adityanair07
Account Settings
Upgrade
Sign out
Your running containers show up here
A container is an isolated environment for your code
What is a container?
5 mins
How do I run a container?
6 mins
View more in the Learning center
Engine running
RAM 1.75 GB CPU 0.50%
Signed in
v4.28.0

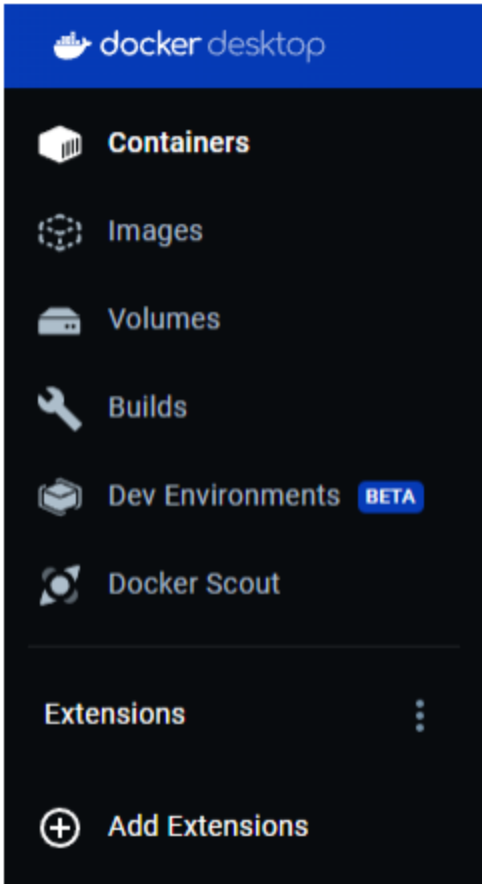
Experiment -4

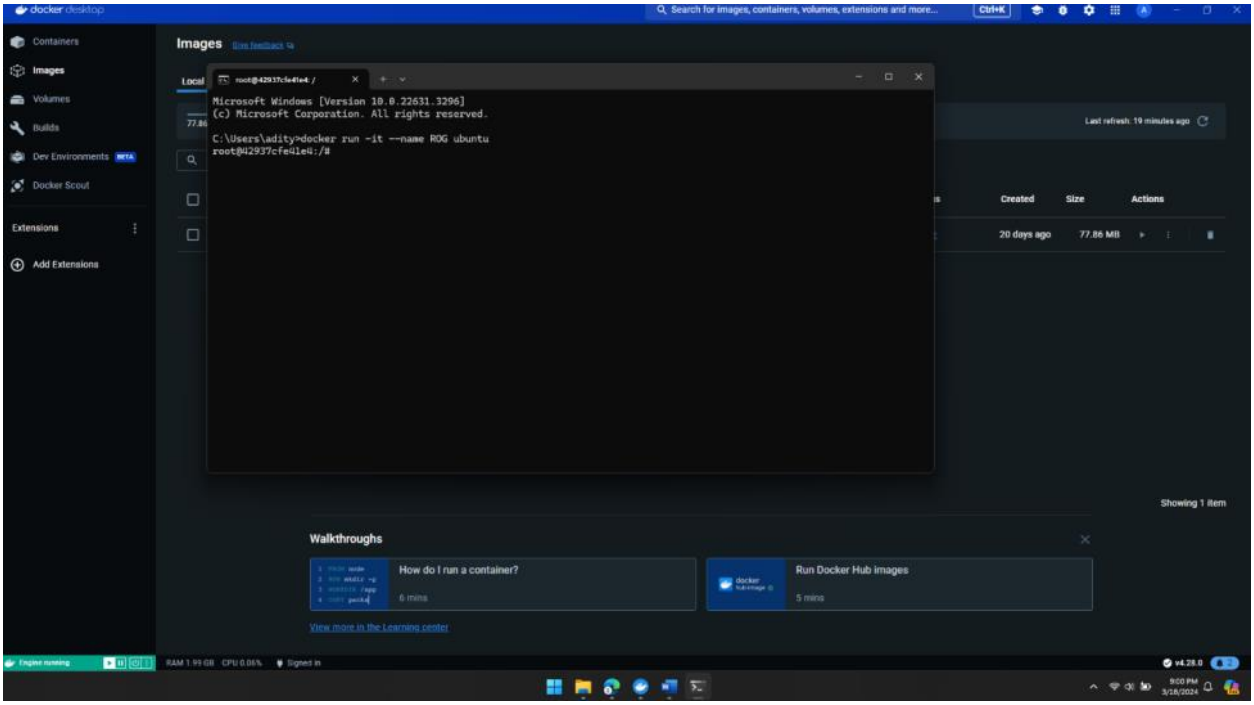
1. Open the docker desktop

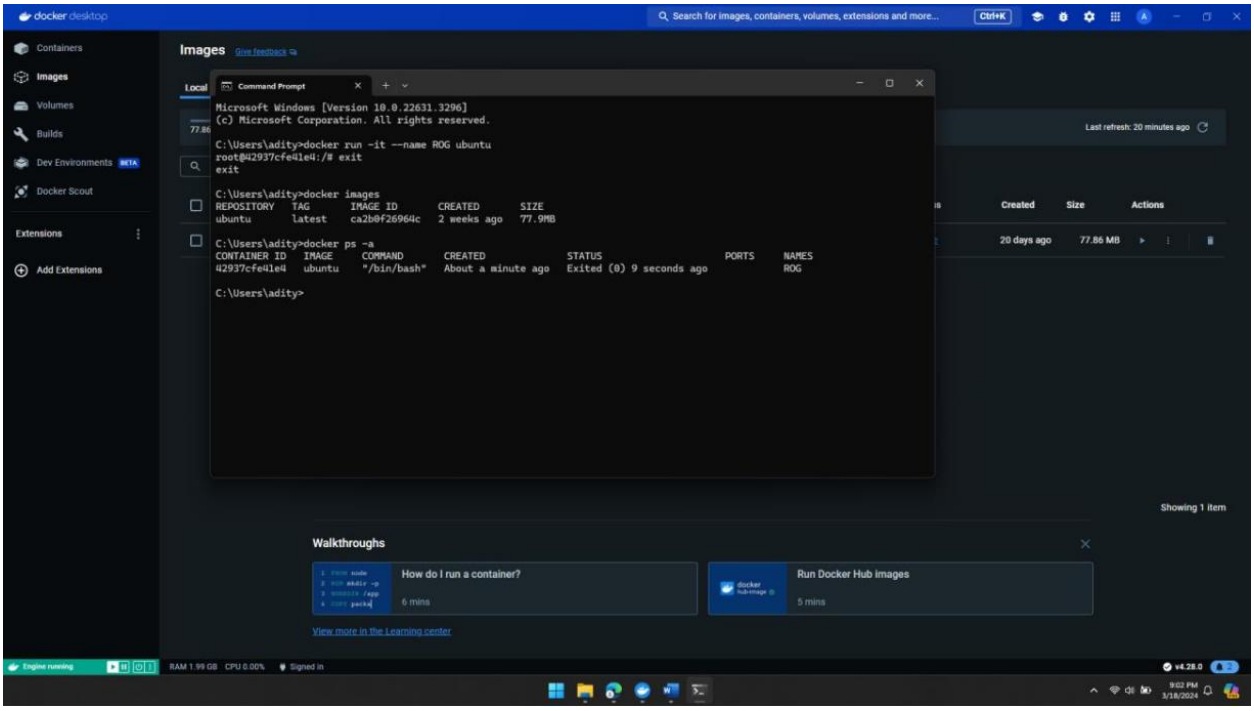
2. Sign into docker desktop via google






docker desktop
Containers
Images
Volumes
Builds
Dev Environments
BETA
Docker Scout
Extensions
Add Extensions


docker desktop
Search for images, containers, volumes, extensions and more...
Ctrl+K
Containers
Images
Volumes
Builds
Dev Environments
BETA
Docker Scout
Extensions
Add Extensions
Images
Give feedback
Local
root@42937cfe41e4: /
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.
C:\Users\adity>docker run -it --name ROG ubuntu
root@42937cfe41e4:/#
Last refresh: 19 minutes ago
Created
Size
Actions
20 days ago
77.86 MB
Showing 1 item
Walkthroughs
How do I run a container?
6 mins
Run Docker Hub images
5 mins
View more in the Learning center
Engine running
RAM 1.99 GB CPU 0.06%
Signed in
v4.28.0


docker desktop
Search for images, containers, volumes, extensions and more...
Ctrl+K
Containers
Images
Volumes
Builds
Dev Environments BETA
Docker Scout
Extensions
Add Extensions
Images
Give feedback
Local
Command Prompt
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.
C:\Users\adity>docker run -it --name ROG ubuntu
root@42937cfe41e4:/# exit
exit
C:\Users\adity>docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest ca2b0f26964c 2 weeks ago 77.9MB
C:\Users\adity>docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
42937cfe41e4 ubuntu "/bin/bash" About a minute ago Exited (0) 9 seconds ago ROG
C:\Users\adity>
Last refresh: 20 minutes ago
Created
Size
Actions
20 days ago
77.86 MB
Showing 1 item
Walkthroughs
How do I run a container?
6 mins
Run Docker Hub images
5 mins
View more in the Learning center
Engine running
RAM 1.99 GB CPU 0.00%
Signed in
v4.28.0

Experiment -5

Using command line, test and validate the functioning

of docker using some basic commands: (a) docker version (b) docker help (c) docker info

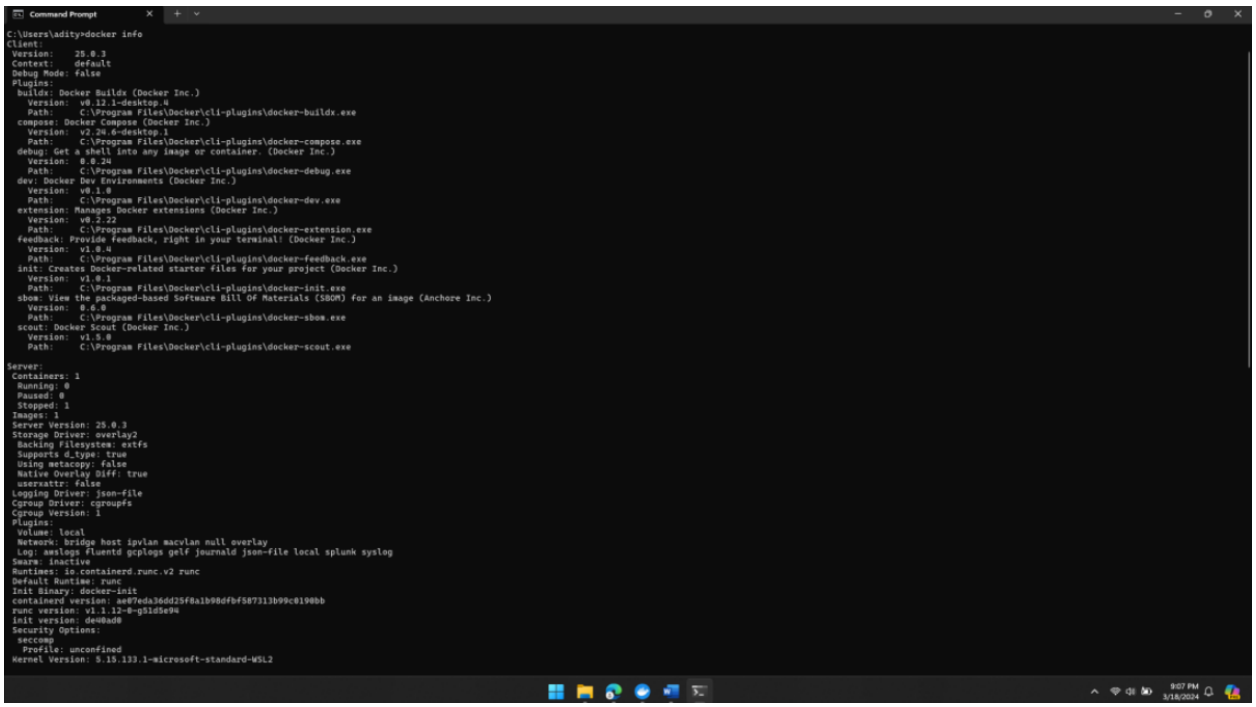
a) docker version

```
Command Prompt

Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\adity>docker version
Client:
 Cloud integration: v1.0.35+desktop.11
 Version:           25.0.3
 API version:       1.44
 Go version:        go1.21.6
 Git commit:        4debf41
 Built:             Tue Feb  6 21:13:02 2024
 OS/Arch:           windows/amd64
 Context:           default

Server: Docker Desktop 4.28.0 (139021)
 Engine:
  Version:          25.0.3
  API version:      1.44 (minimum version 1.24)
  Go version:       go1.21.6
  Git commit:       f417435
  Built:            Tue Feb  6 21:14:25 2024
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.28
  GitCommit:        ae07eda36dd25f8a1b98dfbf587313b99c0190bb
 runc:
  Version:          1.1.12
  GitCommit:        v1.1.12-0-g51d5e94
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0

C:\Users\adity>
```

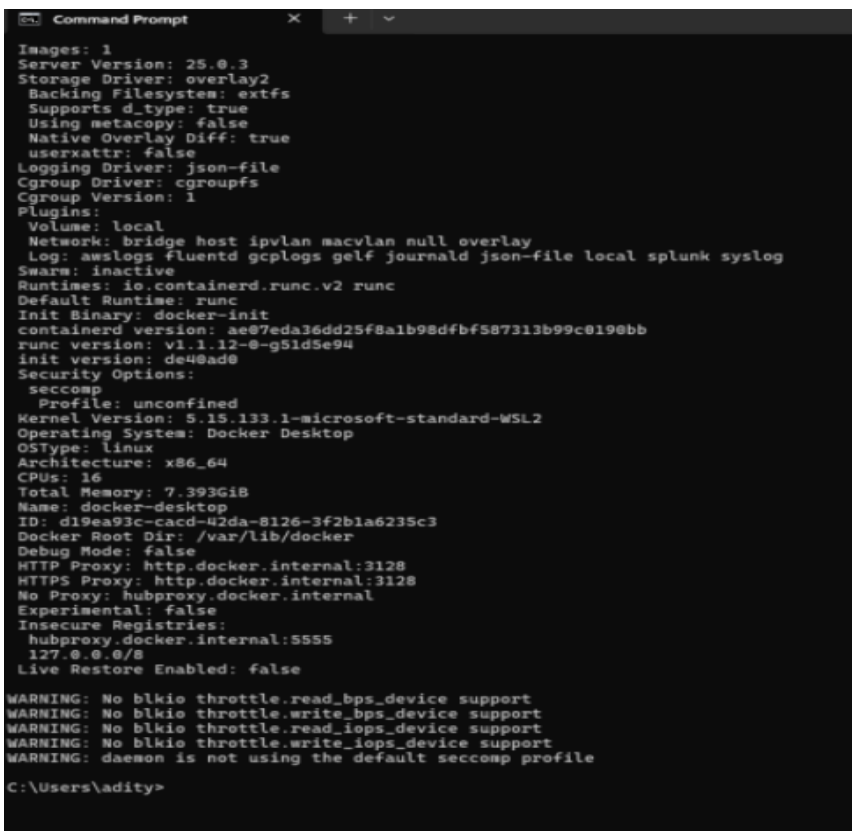
b) docker help

```
Command Prompt

C:\Users\adity>docker help

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
  run         Create and run a new container from an image
  exec        Execute a command in a running container
  ps          List containers
  build       Build an image from a Dockerfile
  pull        Download an image from a registry
  push        Upload an image to a registry
  images      List images
  login       Log in to a registry
  logout      Log out from a registry
  search      Search Docker Hub for images
  version     Show the Docker version information
  info        Display system-wide information

Management Commands:
  builder     Manage builds
  buildx*     Docker Buildx (Docker Inc., v0.12.1-desktop.4)
  compose*    Docker Compose (Docker Inc., v2.24.6-desktop.1)
  container   Manage containers
  context     Manage contexts
  debug*      Get a shell into any image or container. (Docker Inc., 0.0.24)
  dev*        Docker Dev Environments (Docker Inc., v0.1.0)
  extension*  Manages Docker extensions (Docker Inc., v0.2.22)
  feedback*   Provide feedback, right in your terminal! (Docker Inc., v1.0.4)
  image       Manage images
  init*       Creates Docker-related starter files for your project (Docker Inc., v1.0.1)
  manifest    Manage Docker image manifests and manifest lists
  network     Manage networks
  plugin      Manage plugins
  sbom*       View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc., 0.6.0)
  scout*      Docker Scout (Docker Inc., v1.5.0)
  system      Manage Docker
  trust       Manage trust on Docker images
  volume      Manage volumes

Swarm Commands:
  swarm       Manage Swarm

Commands:
  attach      Attach local standard input, output, and error streams to a running container
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's filesystem
  events      Get real time events from the server
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  import      Import the contents from a tarball to create a filesystem image
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  logs        Fetch the logs of a container
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  rmi         Remove one or more images
  save        Save one or more images to a tar archive (streamed to STDOUT by default)
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top         Display the running processes of a container
```

c) docker info



```
C:\Users\adity>docker info
Client:
 Version:    25.0.3
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version:  v0.12.1-desktop.4
    Path:     C:\Program Files\Docker\cli-plugins\docker-buildx.exe
  compose: Docker Compose (Docker Inc.)
    Version:  v2.24.6-desktop.1
    Path:     C:\Program Files\Docker\cli-plugins\docker-compose.exe
  debug: Get a shell into any image or container. (Docker Inc.)
    Version:  0.0.24
    Path:     C:\Program Files\Docker\cli-plugins\docker-debug.exe
  dev: Docker Dev Environments (Docker Inc.)
    Version:  v0.1.0
    Path:     C:\Program Files\Docker\cli-plugins\docker-dev.exe
  extension: Manages Docker extensions (Docker Inc.)
    Version:  v0.2.22
    Path:     C:\Program Files\Docker\cli-plugins\docker-extension.exe
  feedback: Provide feedback, right in your terminal! (Docker Inc.)
    Version:  v1.0.4
    Path:     C:\Program Files\Docker\cli-plugins\docker-feedback.exe
  init: Creates Docker-related starter files for your project (Docker Inc.)
    Version:  v1.0.1
    Path:     C:\Program Files\Docker\cli-plugins\docker-init.exe
  sbom: View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc.)
    Version:  0.6.0
    Path:     C:\Program Files\Docker\cli-plugins\docker-sbom.exe
  scout: Docker Scout (Docker Inc.)
    Version:  v1.5.0
    Path:     C:\Program Files\Docker\cli-plugins\docker-scout.exe

Server:
 Containers: 1
  Running: 0
  Paused: 0
  Stopped: 1
 Images: 1
 Server Version: 25.0.3
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Using metacopy: false
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: ae07eda36dd25f8a1b98dfbf587313b99c0190bb
 runc version: v1.1.12-0-g51d5e94
 init version: de40ad0
 Security Options:
  seccomp
   Profile: unconfined
 Kernel Version: 5.15.133.1-microsoft-standard-WSL2
```



```
 Images: 1
 Server Version: 25.0.3
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Using metacopy: false
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: ae07eda36dd25f8a1b98dfbf587313b99c0190bb
 runc version: v1.1.12-0-g51d5e94
 init version: de40ad0
 Security Options:
  seccomp
   Profile: unconfined
 Kernel Version: 5.15.133.1-microsoft-standard-WSL2
 Operating System: Docker Desktop
 OSType: linux
 Architecture: x86_64
 CPUs: 16
 Total Memory: 7.393GiB
 Name: docker-desktop
 ID: d19ea93c-cacd-42da-8126-3f2b1a6235c3
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 HTTP Proxy: http.docker.internal:3128
 HTTPS Proxy: http.docker.internal:3128
 No Proxy: hubproxy.docker.internal
 Experimental: false
 Insecure Registries:
  hubproxy.docker.internal:5555
  127.0.0.0/8
 Live Restore Enabled: false

WARNING: No blkio throttle.read_bps_device support
WARNING: No blkio throttle.write_bps_device support
WARNING: No blkio throttle.read_iops_device support
WARNING: No blkio throttle.write_iops_device support
WARNING: daemon is not using the default seccomp profile

C:\Users\adity>
```

Experiment - 6

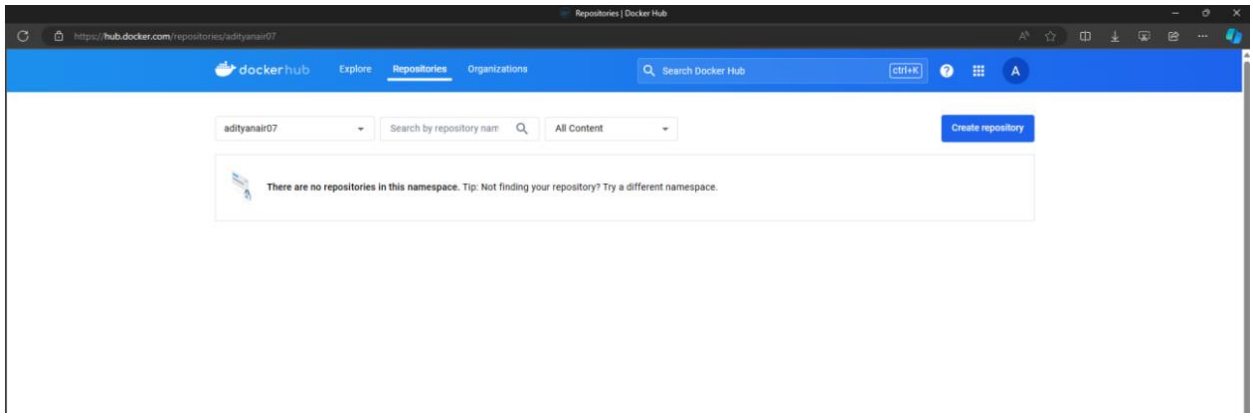
As a client, pull and run default docker image ‘hello□world’ on CLI. Once done, check the following: a. status of container formed at runtime using ‘docker info’ command b. Get details of image id, size and images

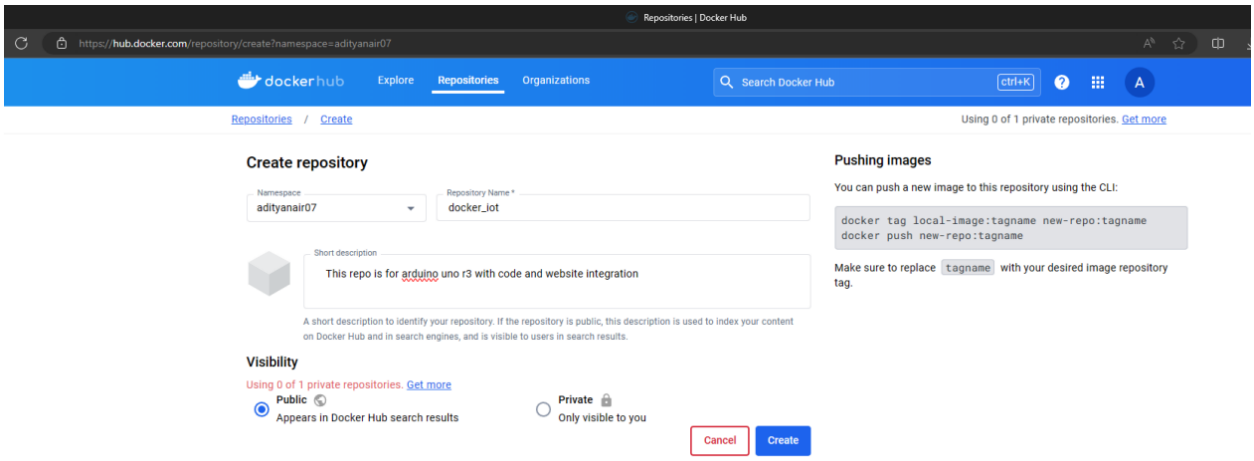
```
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\adity>docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:6352af1ab4ba4b138648f8ee88e63331aae519946d3b67dae50c313c6fc8200f
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview hello-world

C:\Users\adity>
```

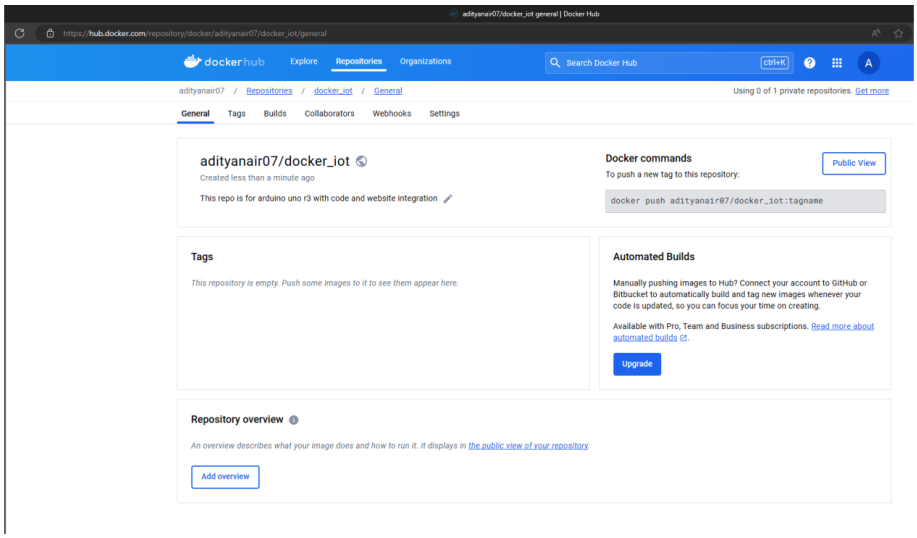
```
C:\Users\adity>docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:6352af1ab4ba4b138648f8ee88e63331aae519946d3b67dae50c313c6fc8200f
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview hello-world

C:\Users\adity>docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/


C:\Users\adity>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

Experiment 7

Log into Docker Hub account with your docker id and create a public repository.






adityanair07/docker_iot:general | Docker Hub
https://hub.docker.com/repository/docker/adityanair07/docker_iot/general
dockerhub
Explore
Repositories
Organizations
Search Docker Hub
ctrl+K
A
adityanair07 / Repositories / docker_iot / General
Using 0 of 1 private repositories. Get more
General
Tags
Builds
Collaborators
Webhooks
Settings
adityanair07/docker_iot
Created less than a minute ago
This repo is for arduino uno r3 with code and website integration
Docker commands
To push a new tag to this repository:
Public View
docker push adityanair07/docker_iot:tagname
Tags
This repository is empty. Push some images to it to see them appear here.
Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions. Read more about automated builds.
Upgrade
Repository overview
An overview describes what your image does and how to run it. It displays in the public view of your repository.
Add overview

Experiment -8

Title: Explore kubernetes:

a) GKE on google cloud platform

b) EKS on AWS.

Theory: Google Kubernetes Engine (GKE):

Google Kubernetes Engine (GKE) is a managed Kubernetes service provided by Google CloudPlatform (GCP). It allows users to deploy, manage, and scale containerized applications using

Kubernetes without the need to manage the underlying infrastructure. Some key features of GKE include:

1. Managed Kubernetes Control Plane: GKE manages the Kubernetes control plane, including master nodes, etcd storage, and control plane components, ensuring high availability and reliability.
2. Node Management: GKE allows users to create and manage clusters of virtual machine instances (nodes) to run their containerized workloads. GKE automatically provisions, upgrades, scales, and repairs these nodes.
3. Integration with Google Cloud Services: GKE integrates seamlessly with other Google Cloud services such as Cloud Logging, Cloud Monitoring, Cloud IAM, and more, making it easier to build, deploy, and manage applications on GCP.
4. Security and Compliance: GKE provides built-in security features such as identity and access management (IAM), network policies, encryption at rest and in transit, and compliance certifications to ensure the security and compliance of containerized workloads.

5. Auto-scaling: GKE supports horizontal pod autoscaling (HPA) and cluster autoscaling, allowing users to automatically scale their applications based on CPU utilization, memory usage, or custom metrics. Amazon Elastic Kubernetes Service (EKS):Amazon Elastic Kubernetes Service (EKS) is a managed Kubernetes service provided by Amazon Web Services (AWS). It enables users to deploy, manage, and scale containerized applications using Kubernetes on AWS infrastructure. Some key features of EKS include:

1. Managed Kubernetes Control Plane: EKS manages the Kubernetes control plane, including master nodes, etcd storage, and control plane components, ensuring high availability and reliability.
2. Node Management: EKS allows users to create and manage clusters of Amazon Elastic Compute Cloud (EC2) instances (nodes) to run their containerized workloads. Users have full control over the EC2 instances and can customize them as needed.
3. Integration with AWS Services: EKS integrates with other AWS services such as Amazon Elastic Block Store (EBS), Amazon Virtual Private Cloud (VPC), AWS Identity and Access Management (IAM), and more, making it easier to build, deploy, and manage applications on AWS.
4. Security and Compliance: EKS provides built-in security features such as IAM integration, network policies, encryption at rest and in transit, and compliance certifications to ensure the security and compliance of containerized workloads.
5. Auto-scaling: EKS supports horizontal pod autoscaling (HPA) and cluster

autoscaling, allowing users to automatically scale their applications based on CPU utilization, memory usage, or custom metrics. Both GKE and EKS offer similar capabilities for running Kubernetes workloads on their respective cloud platforms, and the choice between them often depends on factors such as familiarity with the platform, existing infrastructure, pricing, and specific requirements of the application