Institute of Computer Technology

B. Tech Computer Science and Engineering

Sub: Identity and Access Management (2CSE507)

Name - Nisarg Prajapati

Branch - Cyber Security

Enrollment no. – 23162171019

Semester - 5

Class - A

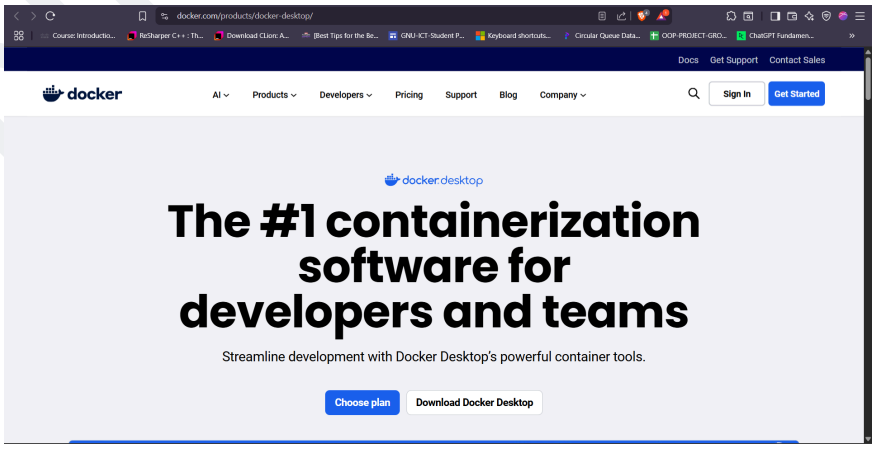
Batch – 52

PRACTICAL NO:- 1

Installing Docker:

1. Download Docker from below given URL:

   https://www.docker.com/products/docker-desktop/



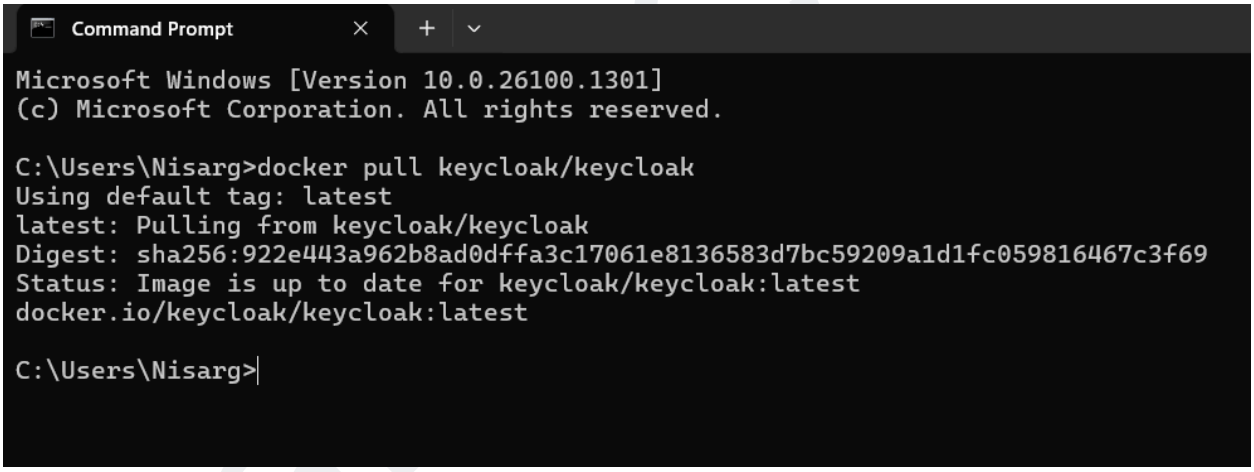
2. After the download completes, Run the installer to install Docker.
3. After installation completes, ‘sign-up’ for an account.

Pulling an image and Creating a Container:

1. Open command prompt (press ‘win+r’, then type ‘cmd’ and press ‘enter’).
2. Then type following command:

    docker pull keycloak/keycloak

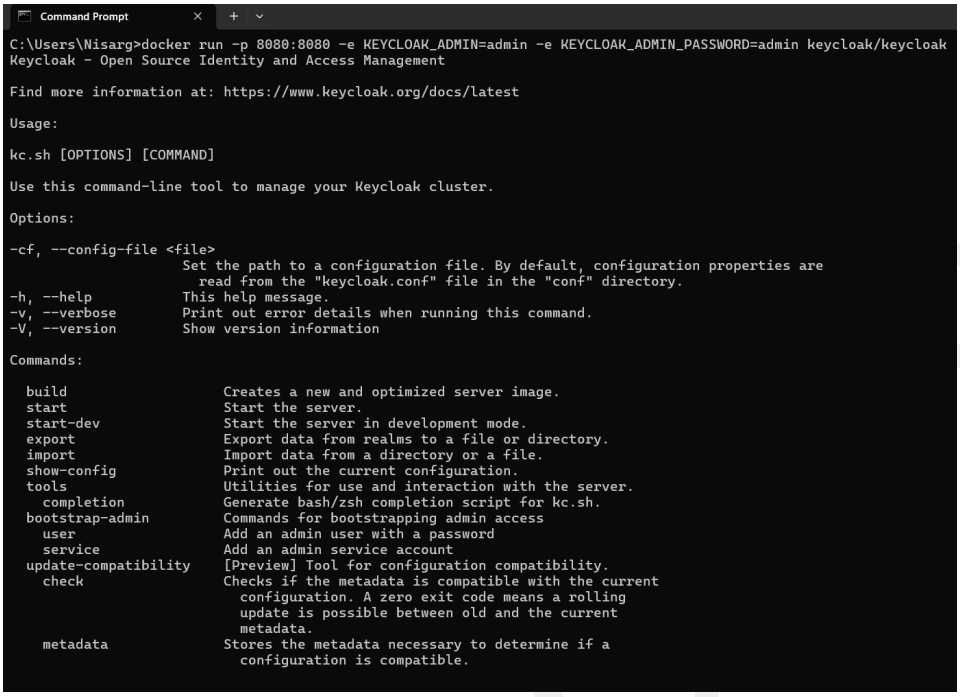
    Explanation: The command docker pull **‘keycloak/keycloak’** downloads the Keycloak container image from Docker Hub to your local system. This image contains everything needed to run the Keycloak identity and access management server inside a Docker container.

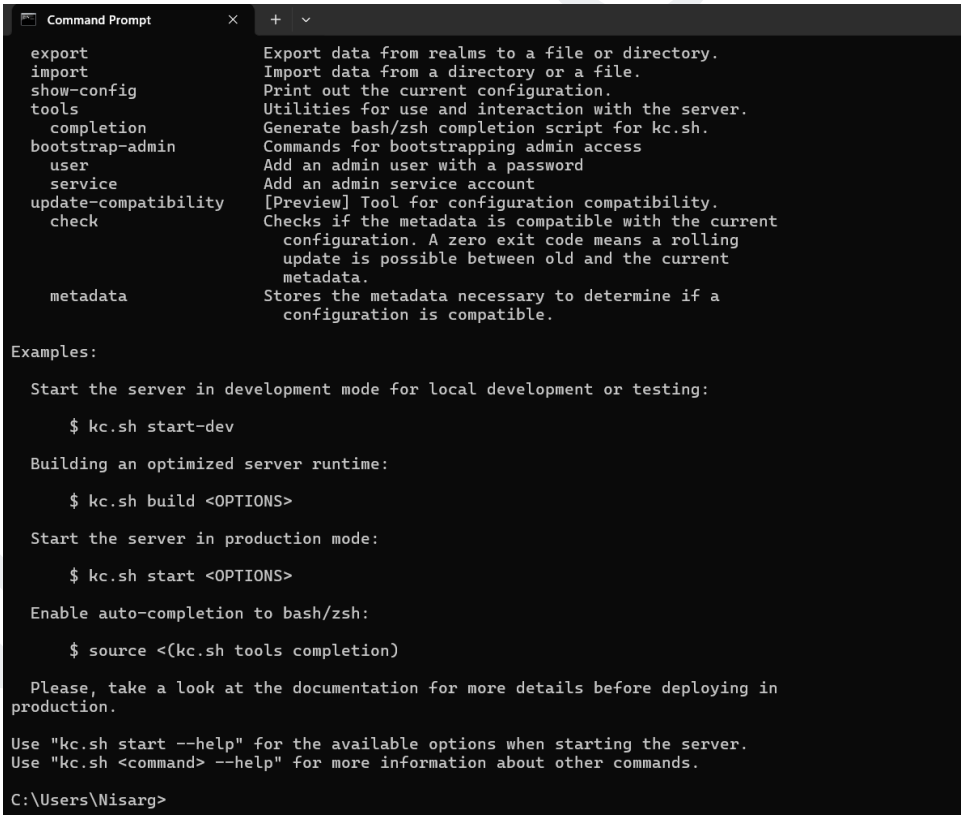


3. Then in order to create an new container, use following command:

    docker run -p 8080:8080 -e KEYCLOAK_ADMIN=admin -e KEYCLOAK_ADMIN_PASSWORD=admin keycloak/keycloak

    Explanation: It runs the Keycloak container and maps host port 8080 to container port 8080. It also sets the environment variables KEYCLOAK_ADMIN and KEYCLOAK_ADMIN_PASSWORD to create the initial admin user (admin/admin). This lets you access the Keycloak admin console via **‘http://localhost:8080’**.

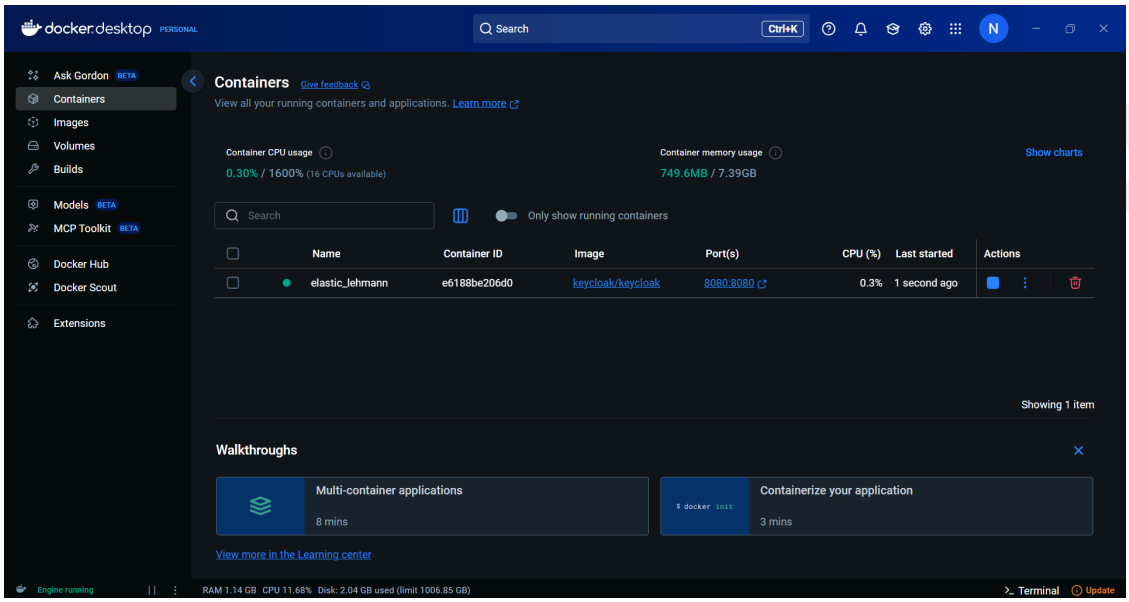




4. Now you will be able to access the container through your Docker Desktop

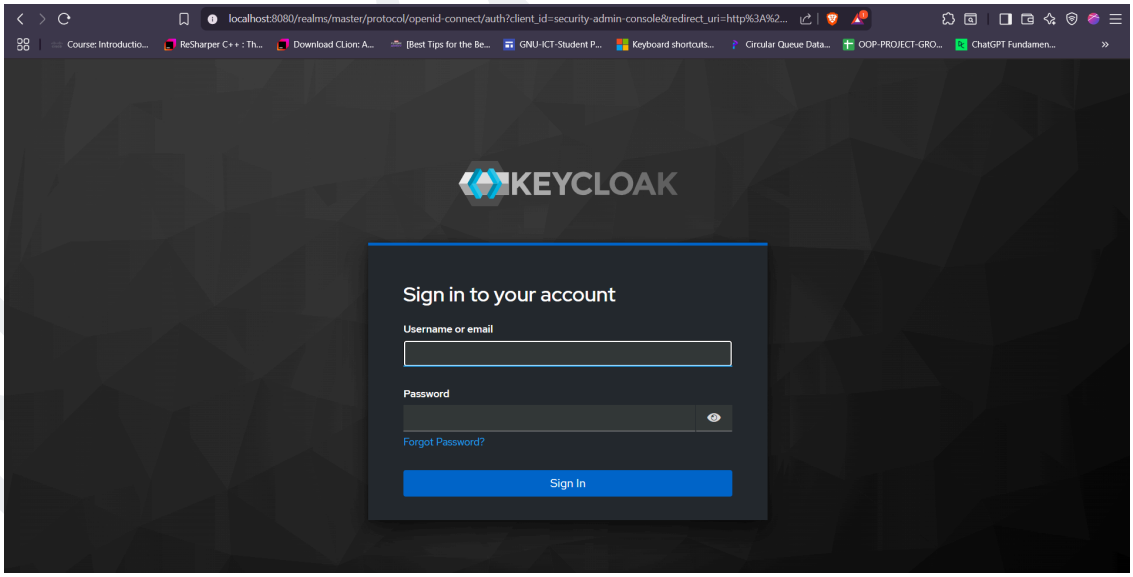
Running a Container:

1. Open your Docker Desktop and go to the container tab.
2. Now, click on the start button of the container you just created.



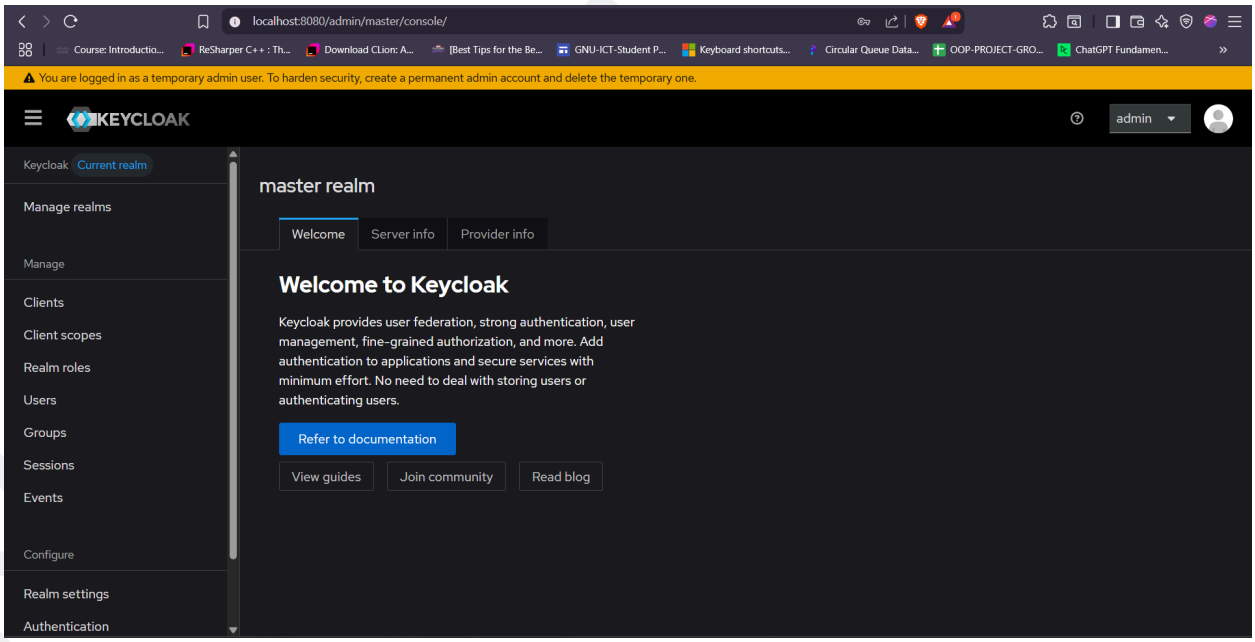
3. Open your browser and type following url:

   ‘localhost:8080’



4. Use the admin login credentials, that you declared during creating the container. (In this case, username: admin and password: admin).

5. Now, you will be able to access admin control of the system.



What is the difference between docker and Virtualization? Why do we use docker?

Features	Docker	Hypervisor
Type	Container-based virtualization	Hardware-based (VM) virtualization
Function	Runs applications in isolated containers	Runs entire OS instances (VMs)
Abstraction Level	OS-level virtualization	Hardware-level virtualization
What It Virtualizes	Only the application and its dependencies	Entire operating system and hardware
Host OS Shared?	Yes (containers share the same host OS kernel)	No (each VM runs its own OS instance)
Guest OS Required	No	Yes
Resource Usage	Lightweight (no guest OS overhead)	Heavy (each VM needs its own OS)
Performance	Near-native	Slightly lower due to VM overhead
Isolation Level	Process-level (less isolated)	Full OS-level (more isolated)
Security	Shared kernel (risk if compromised)	Better security due to full isolation
Portability	Very portable (container images)	Less portable (VM images are larger)
Typical Use Case	Microservices, DevOps, CI/CD, lightweight apps	Full OS testing, legacy app support, multiple OS |