

EXPERIMENT-06

AIM: Explore Docker commands for content management.

Objective: To explore and understand Docker commands for managing images, containers, files, and volumes (content management).

Software Requirements

- **Docker installed and configured** (Docker Desktop for Windows/macOS or Docker Engine for Linux) – <https://www.docker.com/get-started>
 - **Operating System:** Linux (Ubuntu 20.04 or later) / Windows 10 or later with WSL2 / macOS (latest)
 - **Command-line terminal** (Bash / PowerShell) for executing Docker commands
 - **Internet connectivity** to pull images from Docker Hub or other registries
 - **Optional text editor** (VS Code) to create and edit files inside containers
-

Docker is an open-source platform that allows developers to easily build, deploy, and run applications in containers.

Docker allows developers to create a container image that includes all the dependencies needed for an application to run, regardless of the underlying operating system or infrastructure.

Docker also provides tools for managing container images, such as Docker Hub, a public registry of images, and Docker Compose, a tool for defining and running multi-container applications.

Docker Commands

1. **docker:** This command is used to list all docker commands.
2. **docker –version:** This command is used to check docker version.
3. **docker pull ImageName:** This command is used to download a Docker image from a registry or from docker hub.
4. **docker run ImageName:** This command is used to create a new Docker container from an image.
 - i. **docker run –d ImageName:** This command is used to build a container in background.
 - ii. **docker run ImageName: Version:** This command is used to pull and create a specific version of an image.

- iii. **docker run -it ImageName:** This command is used to access terminal of that container in our local machine in the interactive mode.
 - iv. **docker run -name containerName -d ImageName:** This command is used to give custom name to a container.
5. **docker images:** This command is used to list all the Docker images that are currently available on your system.
 6. **docker rm ContainerName:** This command is used to remove a Docker container.
 7. **docker rmi ImageName:** This command is used to remove an docker image.
 8. **docker start ContainerName or ContainerID:** This command is used to restart an existing container.
 9. **docker stop ContainerName or ContainerID:** This command is used to stop a running container.
 10. **docker ps:** This command is used to list all the running Docker containers.
 11. **docker ps -a:** This command is used to list all the Docker containers, including running and stopped containers.
 12. **docker logs ContainerID or ContainerName:** This command is used to check the logs of the container which can help to identify the root cause of the problem.
 13. **docker exec -it ContainerID or ContainerName: bash/terminal/sh**
This command is used to execute additional commands in a running container. We can access that containers bash/terminal.
 14. **docker compose -f FileName.yaml up -d:** This command is used to manage multi-container Docker applications.
 - f: Option for file name
 - up -d: Whenever the containers are defined inside the yaml file and if we want to create and start them in the detached mode we will choose up.
 15. **docker compose -f FileName.yaml down:** Whenever the containers are defined inside the yaml file and if we want to delete them then we will choose down.

16. docker build -t ImageName: version . : This command is used to build docker image.

-t or—tag: Version of images

dot (.): Build context (Current Directory)

17. docker login -u UserName: The Docker login command will help you to authenticate with the Docker hub by which you can push and pull your images.

18.docker push Image Name: This command is used to push docker image to dockerhub.

19.docker volume create VolumeName: This command is used to create a volume.

20.docker volume ls: This command is used to list all volumes available.

21.Docker run -it -v VolumeName: ContainerPath: This command is used to attach the volume to a path inside the container.

22.docker volume rm VolumeName: This command is used to remove the volume.

23.docker volume prune: This command is used to remove unused volumes.

Some important flags

-d: To run any container in background.

-e: To set environment variables

-p: To bind the host port with the container port