

|  |  |
| --- | --- |
| **Project Name:** | Docker |
| **Members:** | Naqqash Haider, Mussab Bin Shahid |
| **Sap-Ids:** | 1980, 2024 |

**Project Introduction:**

Docker is an open-source platform that allows developers to easily create, deploy, and run applications in containers. Containers are lightweight, standalone environments that include all the files and dependencies needed to run an application. This makes it easy to deploy and run applications in different environments, such as in development, testing, and production.

Docker uses containerization technology, which allows it to package an application and its dependencies into a single container. This container can then be run on any machine that supports Docker, ensuring that the application runs consistently across different environments.

Docker also uses a Docker daemon, which is responsible for managing the containers on a host machine. Developers can use the Docker command-line interface (CLI) or a graphical user interface (GUI) to interact with the daemon and manage their containers. Docker also has a public registry called Docker Hub, which allows developers to share and distribute their images.

Docker provides many benefits, including increased portability, consistency, and isolation, as well as improved performance and scalability. It has become a widely adopted technology in the software development industry and is used by many organizations to improve the efficiency and reliability of their applications.

**How it works:**

* Docker allows developers to package and distribute software in the form of containers.
* Containers are lightweight and portable environments that include all the files and dependencies needed to run an application.
* Docker uses a Docker daemon to manage the containers on a host machine.
* Developers can use the Docker CLI or GUI to interact with the daemon and manage their containers.
* Containers are created based on pre-configured images and can be shared through a public registry called Docker Hub.
* Docker provides consistency, isolation, and portability across different environments.
* Docker facilitates scaling and versioning of applications.

**Modules description:**

1. **Basic Use of Docker on Ubuntu Terminal:**

Here we have demonstrated some basic docker commands such as

docker pull <repository name>: By using this command we can pull any repository from docker hub to our local machine such as docker pull ubuntu.

docker ps: shows number of processes or containers running in our local machine in our local machine.

docker rmi: for removing images.

docker rm: for removing containers.

1. **Making Django web app using docker**
2. **Dockerizing existing Django web app.**
3. **Pushing Docker container on Docker Hub.**

https://hub.docker.com/r/hydrabotquality2cvz/hydrabotwebsite-master\_web

**Advantages**

* Consistent environment: Docker provides consistency across all environments, from development to production, by packaging the application and its dependencies together in a container.
* Lightweight: Containers are lightweight and fast, as they share the host machine's kernel and do not require a full operating system.
* Portable: Containers can run on any machine that has Docker installed, regardless of the underlying operating system or system configuration.
* Scalable: Docker allows you to easily scale an application by running multiple instances of a container.
* Isolation: Containers are isolated from the host machine and from each other, which improves security and reduces conflicts.
* Versioning: Applications can be versioned, which makes it easy to roll back to a previous version of the application, in case of issues.
* Cost-effective: Docker allows us to use the same infrastructure and configuration for different environments, which can help to reduce costs.

**Main Functionality**

Docker's main functionality is to provide a way to package and distribute software in the form of containers. Containers let developers package an application with all its dependencies, such as libraries and system tools, and run it consistently on any machine that supports Docker. This makes it easy to deploy and run applications in different environments, such as in development, testing, and production.

**Tools, Technologies, and Platform used:**

**Tools:** VS code

**Technologies:** Python, Django, HTML & CSS, JavaScript and Docker

**Platform:** Ubuntu 20 LTS (Linux)

**Personal contribution and your role in the project:**

**Slides and Presentation:** Both

**Docker Practical:** Both

**Front End:** Naqqash

**Backend:** Mussab

**Challenges in the project:**

When "dockerizing" a Django web application, some common challenges that may be faced include:

* Configuring and connecting the different components of the application inside the container.
* Managing dependencies and environment-specific configurations.
* Handling persistent data and debugging issues inside the container.
* Optimizing container configuration.
* Managing container images and dependencies.
* Monitoring and scaling the app in containerized environment.
* Deployment and rollback of the containers.
* Compliance with host machine security policies.

**The number of people in the project:**

There are 2 people in the project.