Experiment 8

Aim : To study and Implement Containerization using Docker

Theory

Containerization is a popular approach to application deployment that allows applications to be packaged and run in a lightweight and isolated environment. Docker is a widely used platform for containerization that provides a set of tools and technologies for building, deploying, and managing containerized applications.

At its core, Docker is based on the concept of a container, which is an isolated environment that encapsulates an application and all of its dependencies. Each container runs its own operating system, providing a level of isolation and security that is not possible with traditional virtual machines. Containers can be easily moved between environments, making them ideal for modern, cloud-based applications that require rapid deployment and scaling.

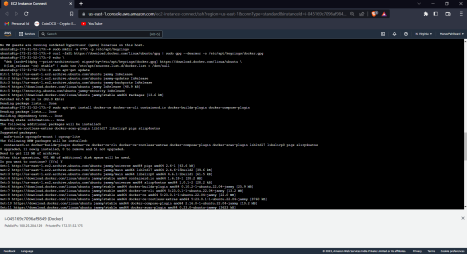
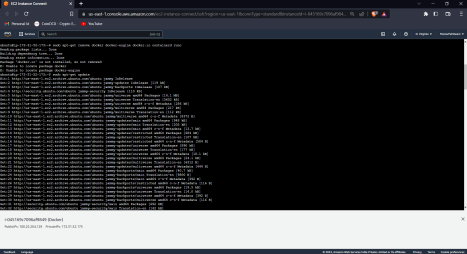
Docker consists of several key components, including the Docker engine, which is responsible for building and running containers, and Docker Hub, which is a central repository for Docker images. Docker images are preconfigured containers that can be used as a starting point for building containerized applications.

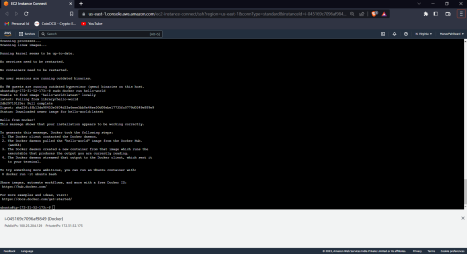
Some key benefits of containerization using Docker include improved application portability, faster deployment times, and better resource utilization. Docker also provides a high degree of flexibility and control, allowing developers to easily manage and scale containerized applications across a variety of environments.



Implementation -

*Installing docker*





*ubuntu@ip-172-31-52-173:~$ sudo docker run hello-world*

*Unable to find image 'hello-world:latest' locally*

*latest: Pulling from library/hello-world*

*2db29710123e: Pull complete*

*Digest:*

*sha256:ffb13da98453e0f04d33a6eee5bb8e46ee50d08ebe17735fc0779d0349e889e9 Status: Downloaded newer image for hello-world:latest*

*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.*

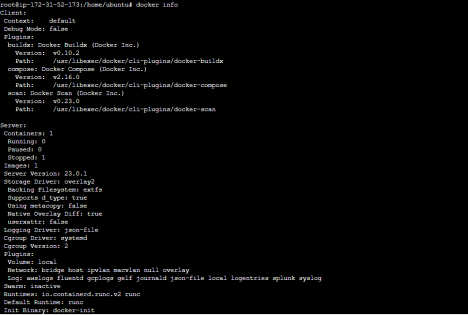
*To try something more ambitious, you can run an Ubuntu container with: $ docker run -it ubuntu bash*

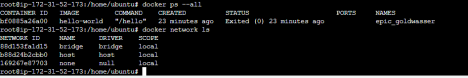
*Share images, automate workflows, and more with a free Docker ID: https://hub.docker.com/*

*For more examples and ideas, visit:*

*https://docs.docker.com/get-started/*

*Exploring Docker Commands :-*





Conclusion

Containerization using Docker is a powerful approach to application deployment that provides a range of benefits for modern, cloud-based applications. By leveraging the tools and technologies provided by Docker, developers can build and deploy containerized applications quickly and easily, while also ensuring a high degree of portability, flexibility, and security.