Experiment No. 9

Aim: To study and Implement Containerization using Docker

Objective: To know the basic differences between Virtual machine and Container. It involves demonstration of creating, finding, building, installing, and running Linux/Windows application containers inside a local machine or cloud platform.

Theory:

- open platform for developing, shipping and running applications
- enables you to separate your applications from your infrastructure so you can deliver software quickly
- you can manage your infrastructure in the same ways you manage your applications
- Docker provides the ability to package and run an application in a loosely isolated environment called a container.
- Containers are lightweight and contain everything needed to run the application, so you do not need to rely on what is currently installed on the host.
- Develop your application and its supporting components using containers.
- The container becomes the unit for distributing and testing your application.
- When you're ready, deploy your application into your production environment, as a container or an orchestrated service. This works the same whether your production environment is a local data center, a cloud provider, or a hybrid of the two.

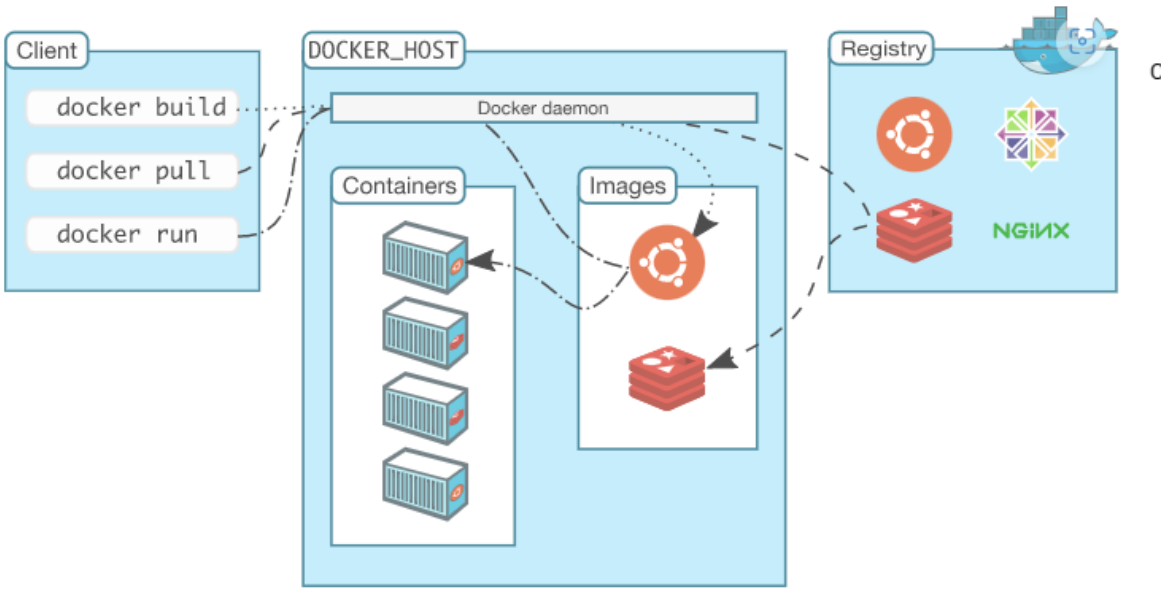
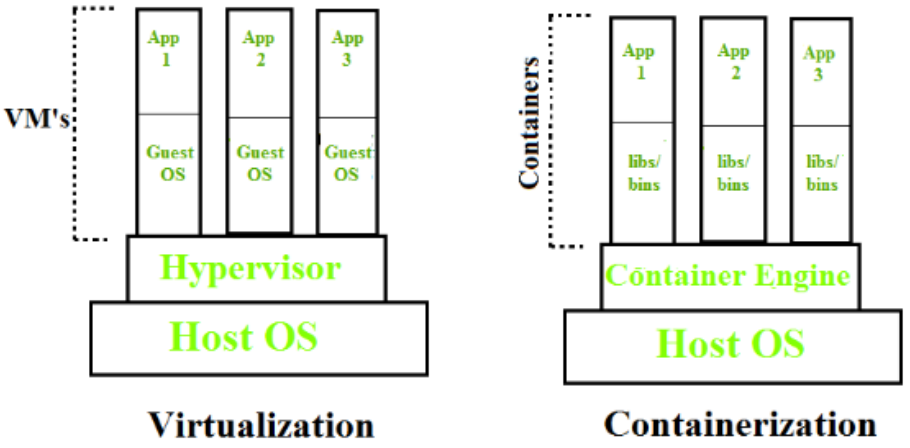


Figure 1: Docker Architecture

Containerization:

- Containerization is OS-based virtualization that creates multiple virtual units in the user space, known as Containers.
- Containers share the same host kernel but are isolated from each other through private namespaces and resource control mechanisms at the OS level.
- Container-based Virtualization provides a different level of abstraction in terms of virtualization and isolation when compared with hypervisors.
- Hypervisors use a lot of hardware which results in overhead in terms of virtualizing hardware and virtual device drivers.
- containers implement isolation of processes at the operating system level, thus avoiding such overhead.
- Containerization has better resource utilization compared to VMs and a short boot-up process. It is the next evolution in virtualization.
- Containers can run virtually anywhere, greatly easy development and deployment: on Linux, Windows, and Mac operating systems; on virtual machines or bare metal, on a developer's machine or in data centers on-premises; and of course, in the public cloud.
- Containers virtualize CPU, memory, storage, and network resources at the OS level, providing developers with a sandboxed view of the OS logically isolated from other applications.
- Docker is the most popular open-source container format available and is supported on Google Cloud Platform and by Google Kubernetes Engine.



Steps:

1. Open docker.com Scroll down, Click on 'Get started for free' tab.
2. Click on Docker Desktop, Download it
3. After downloading, Open 'Docker Desktop Installer' & start installation
4. After Installation, Restart your device.

5. Accept the terms and conditions, Click on Accept
6. Click on the link - https://aka.ms/wsl2kernel. (Do not close this window).
7. Download the WSL2 Linux kernel update package for x64 machines.
8. A ft e r D o w nlo a d is c o m ple t e, R u n t h e . m s i p a c k a g e. Click on next
9. After, the setup is complete, Click on finish.
10. Open Powershell as an Administrator
11. Run the following Command:
    wsl --set-default-version 2
12. Now, Click on Restart
13. Docker should now restart. Click on Start.
14. Open Command Prompt, run the following commands:
    1. To check the version of Docker:
       docker --version
2. To install image of ubuntu
   docker pull ubuntu
3. Check downloaded images
   docker images
4. Run ubuntu OS
   docker run -it ubuntu /bin/bash
5. Open another Command Prompt and follow the steps shown below
   -docker ps

   docker container ls –a

   docker container rm b71e3e6b1118 //copy docker id for remove but first
   (Use your container ID in the above command)

   stop your docker
   - docker container stop b71e3e6b1118
   - docker container rm b71e3e6b1118
   - docker ps
   - docker //list all docker commands
   - docker images
   - docker image rm ff0fea8310f3 // copy image id from previous output
   (Use your image ID in the above command)
   - docker run -it ubuntu /bin/bash //check output

Output/Observation:

Conclusion:

Comment on implementation of Containerization using Docker