# Docker Basics
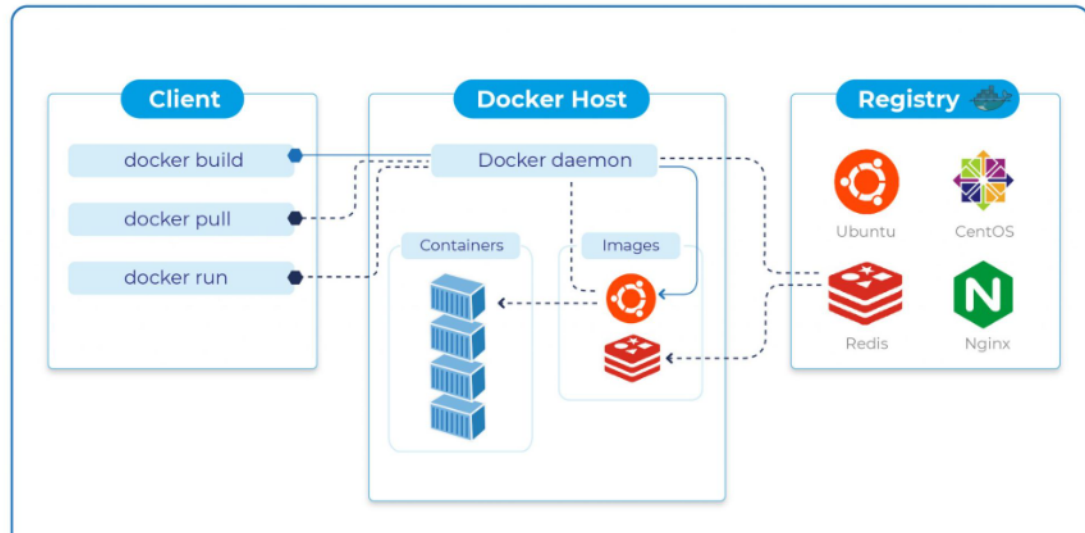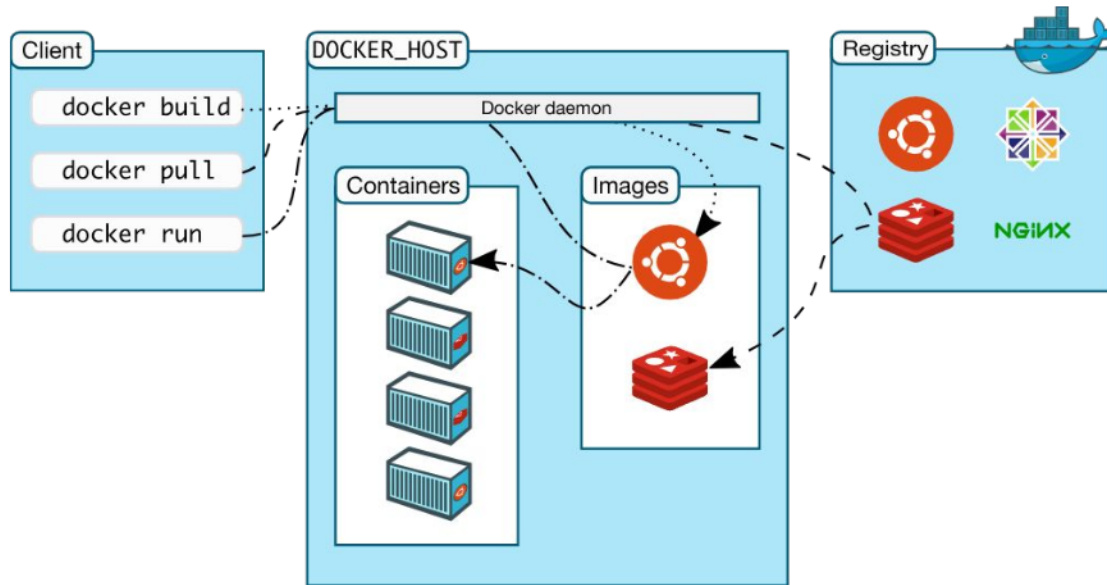
Pramod Dutta

Let's Learn Basics of Docker for QA/ Software Tester.

# Agenda

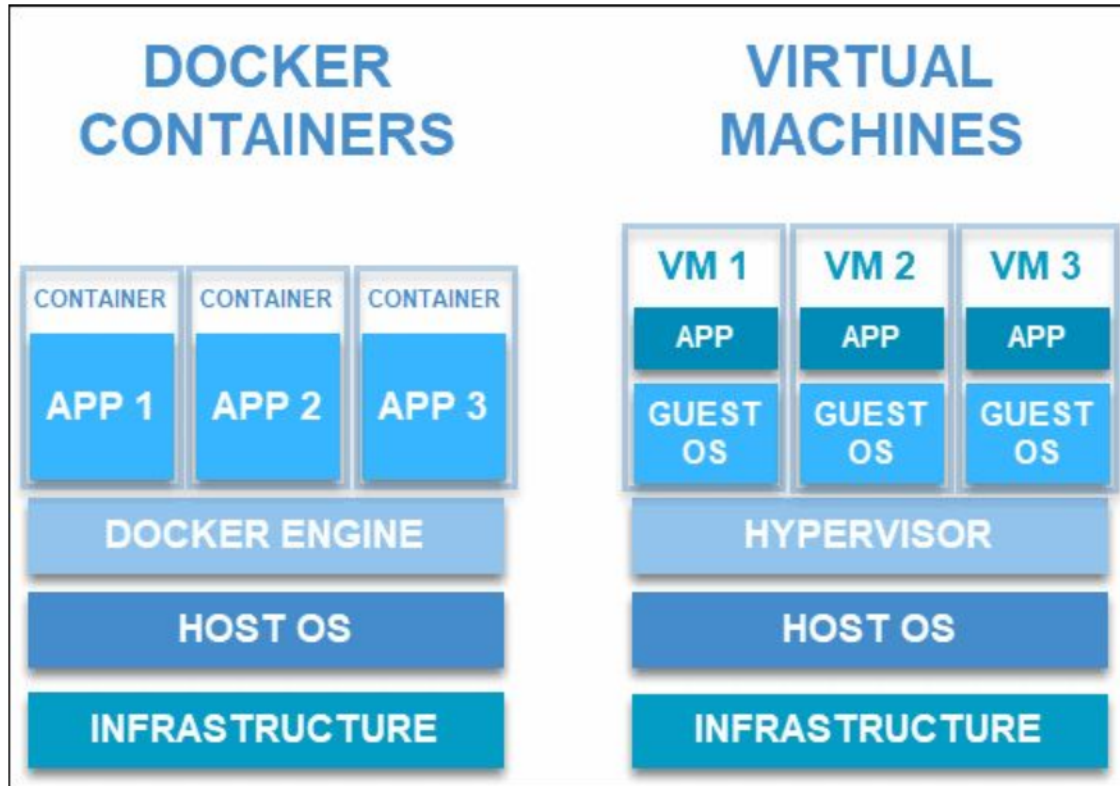- **What is Docker?**

- **Key Concepts of Docker**

- **Installation**

- **Docker Commands**

- **Docker Images**

- **Docker compose**

- **Running Jenkins on Docker.**

- **Running Seelnium Grid.**

# What is Docker?

Docker is a platform for developing, shipping, and running applications in containers. Containers are lightweight, portable, and self-sufficient environments that include everything your application needs to run.
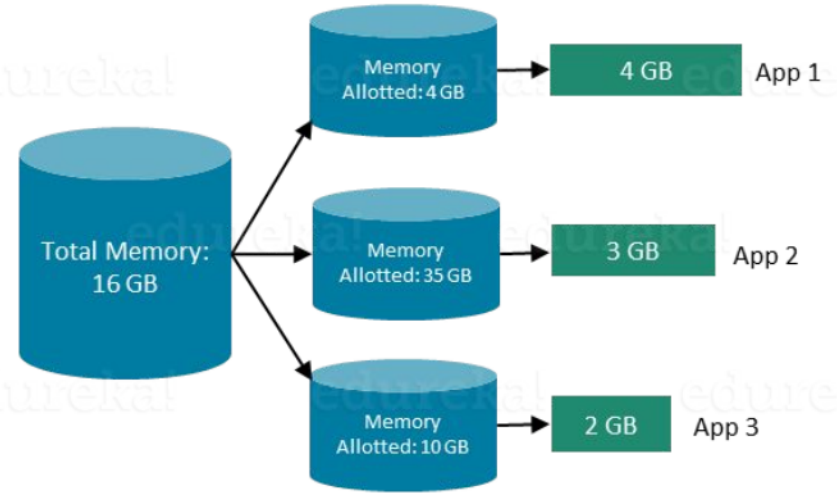
# In case of Virtual Machines



Total Memory: 16 GB

6 GB → 4 GB | 2 GB — VM 1

4 GB → 3 GB | 1 GB — VM 2

6 GB → 2 GB | 4 GB — VM 3

■ → Memory Used: 9 GB

■ → Memory wasted: 7 GB

7 Gb of Memory is blocked and cannot be allotted to a new VM

# In case of Docker

Total Memory: 16 GB

Memory Allotted: 4 GB → 4 GB — App 1

Memory Allotted: 35 GB → 3 GB — App 2

Memory Allotted: 10 GB → 2 GB — App 3

■ → Memory Used: 9 GB

Only 9 GB memory utilized; 7 GB can be allotted to a new Container

# Key Docker Concepts:

Images: Docker images are like blueprints for containers. They contain all the necessary files and configurations to run an application.

Containers: Containers are instances of Docker images. They run applications in isolated environments, ensuring consistency across different environments.

# Docker image vs Docker Container



Docker Image

1GB

Run Image → Container 1 1 GB

Run Image → Container 2 1 GB

Run Image → Container 3 1 GB
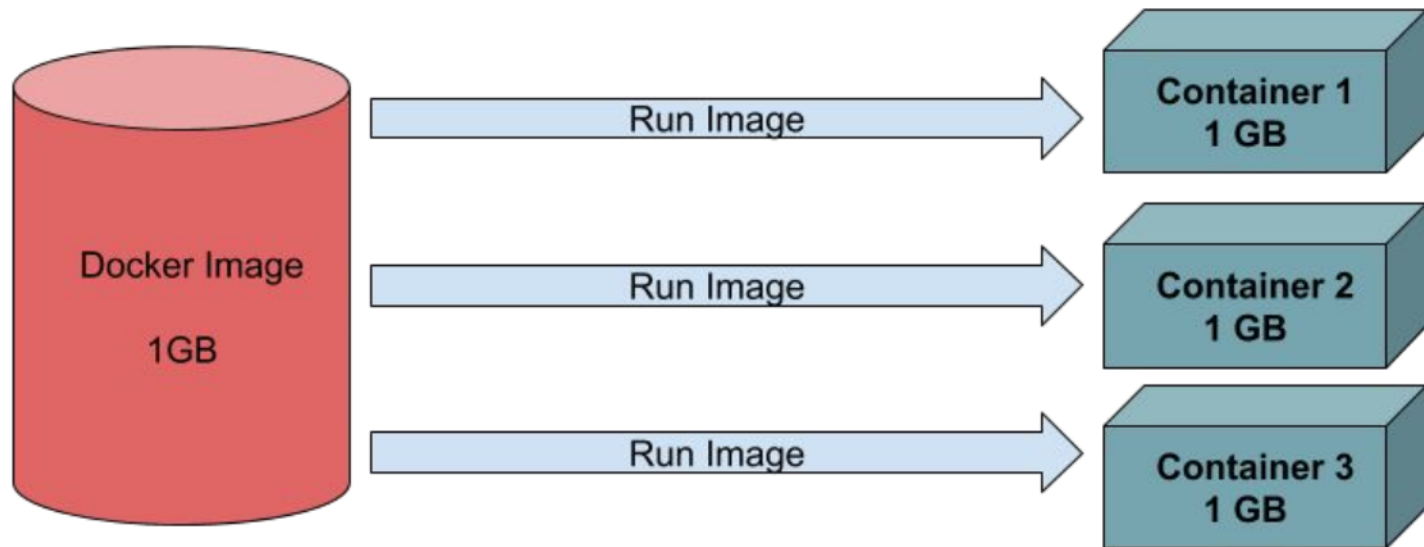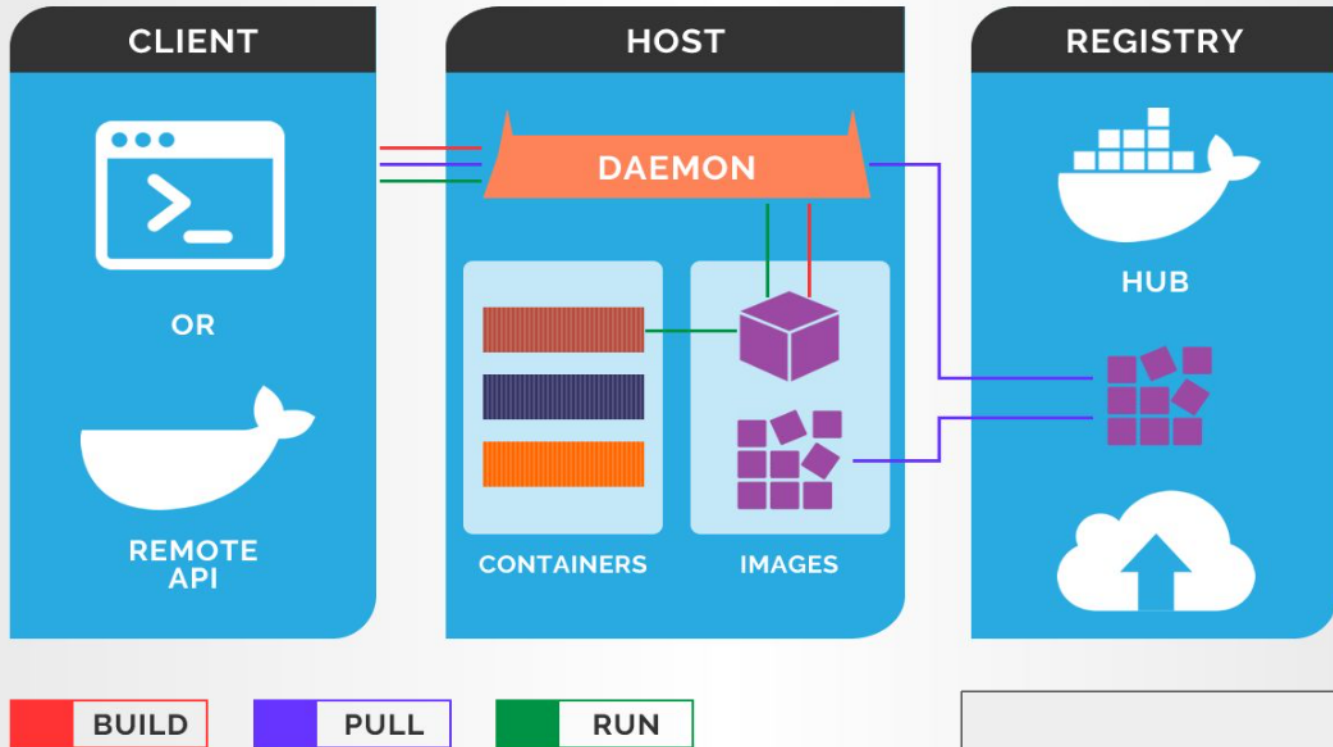
# Installation

Install Docker on your testing machine by following the official installation instructions for your OS (Windows, macOS, or Linux).

https://docs.docker.com/engine/install/

## Desktop

| Platform | x86_64 / amd64 | arm64 (Apple Silicon) |
|---|---|---|
| Docker Desktop for Linux | ✅ | |
| Docker Desktop for Mac (macOS) | ✅ | ✅ |
| Docker Desktop for Windows | ✅ | |

# Basic Docker Commands:

| | | | |
|---|---|---|---|
| `docker ps` | List the running containers. | `docker logs -f --until= [interval] [container]` | Retreive logs before a specific point in time. |
| `docker ps -a` | List all the containers, both running and stopped. | `docker events [container]` | View real time events for a container. |
| `docker create [image]` | Create a container without starting it. | `docker update [container]` | Update the configuration of a container. |
| `docker create -it [image]` | Create an interactive container with pseudo-TTY. | `docker port [container]` | Show port mapping for a container. |
| `docker rename [container] [new-name]` | Rename a container. | `docker top [container]` | Show running processes in a container. |
| `docker rm [container]` | Remove a stopped container. | `docker stats [container]` | Show live resource usage statistics for a container. |
| `docker rm -f [container]` | Force remove a container, even if it is running. | `docker diff [container]` | Show changes to files or directories on the filesystem. |
| `docker logs [container]` | View logs for a running container. | `docker cp [file-path] CONTAINER:[path]` | Copy a local file to a directory in a container. |

# Basic Docker Commands:

| Docker Command | Information |
|---|---|
| ps/container ls | Lists all containers that are currently running on a system & also checks their status with related statistics |
| images/image ls | Lists of all images existing within a Docker system |
| logs | Displays logged information of specific containers |
| version | Details information of the Docker version installed on the system |
| info | Displays system-wide information of the host system where Docker is installed |
| inspect | Displays low level information of both containers and images |
| history | Details history of any image existing on a Docker system |
| port | Reveals host to container port mappings |
| diff | Displays a list of changed files and directories in a container filesystem w.r.t. addition, deletion & change |
| top | Displays a list of processes running within a container |
| stats | Lists container details along-with system resource usage(CPU, RAM, Network I/O, Block I/O, Process IDs) |

```
Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
  run        Create and run a new container from an image
  exec       Execute a command in a running container
  ps         List containers
  build      Build an image from a Dockerfile
  pull       Download an image from a registry
  push       Upload an image to a registry
  images     List images
  login      Log in to a registry
  logout     Log out from a registry
  search     Search Docker Hub for images
  version    Show the Docker version information
  info       Display system-wide information

Management Commands:
  builder      Manage builds
  buildx*      Docker Buildx (Docker Inc., v0.11.2-desktop.4)
  compose*     Docker Compose (Docker Inc., v2.21.0-desktop.1)
  container    Manage containers
  context      Manage contexts
  dev*         Docker Dev Environments (Docker Inc., v0.1.0)
  extension*   Manages Docker extensions (Docker Inc., v0.2.20)
  image        Manage images
  init*        Creates Docker-related starter files for your project (Docker Inc., v0.1.0-beta.1
  manifest     Manage Docker image manifests and manifest lists
  network      Manage networks
  plugin       Manage plugins
  sbom*        View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore
  scan*        Docker Scan (Docker Inc., v0.26.0)
  scout*       Command line tool for Docker Scout (Docker Inc., 0.24.1)
```

```
docker run alpine:latest

docker run alpine:latest echo "Hello World"
```

docker pull ubuntu:20.04

docker run -it ubuntu /bin/bash

```
(base) → ~ docker run -it ubuntu /bin/bash
root@b0eab0fad2fd:/# uname -a
Linux b0eab0fad2fd 6.3.13-linuxkit #1 SMP PREEMPT_DYNAMIC Thu Sep  7 07:54:49 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
root@b0eab0fad2fd:/# 
```
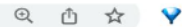
```
docker pull jenkins/jenkins:lts

docker run -d -p 8080:8080 -p 50000:50000 --name
my-jenkins-container jenkins/jenkins:lts

docker logs my-jenkins-container

docker stop my-jenkins-container
docker rm my-jenkins-container

docker exec 6f07f12df3b8 cat
/var/jenkins_home/secrets/initialAdminPassword
```

Please wait while Jenkins is getting ready to work ...

Your browser will reload automatically when Jenkins is ready.

# **Selenium Test on Docker**

https://github.com/SeleniumHQ/docker-selenium

docker run -d -p 4444:4444 -p 7900:7900 --shm-size="2g" selenium/standalone-firefox:4.12.1-20230912

https://dockerlabs.collabnix.com/docker/cheatsheet/