* **Overview**

Docker is a system container manager.

* **Docker Engine**

Docker Engine is a client-server application with these major components:



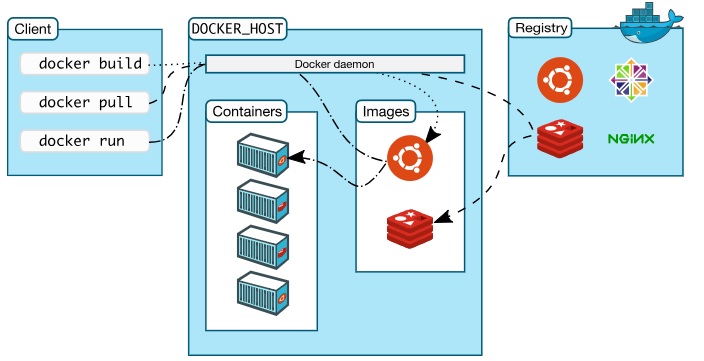
-server which is a type of long-running program called a daemon process (the dockerd command).

-REST API which specifies interfaces that programs can use to talk to the daemon and instruct it what to do.

-command line interface client (the docker command).

* **Docker Architecture**

Docker uses a client-server architecture. The Docker client talks to the Docker daemon, which does the heavy lifting of building, running, and distributing your Docker containers.



**The Docker daemon**

The Docker daemon (dockerd) listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes.

**The Docker client**

The Docker client (docker) is the primary way that many Docker users interact with Docker.

**Docker registries**

A Docker registry stores Docker images.

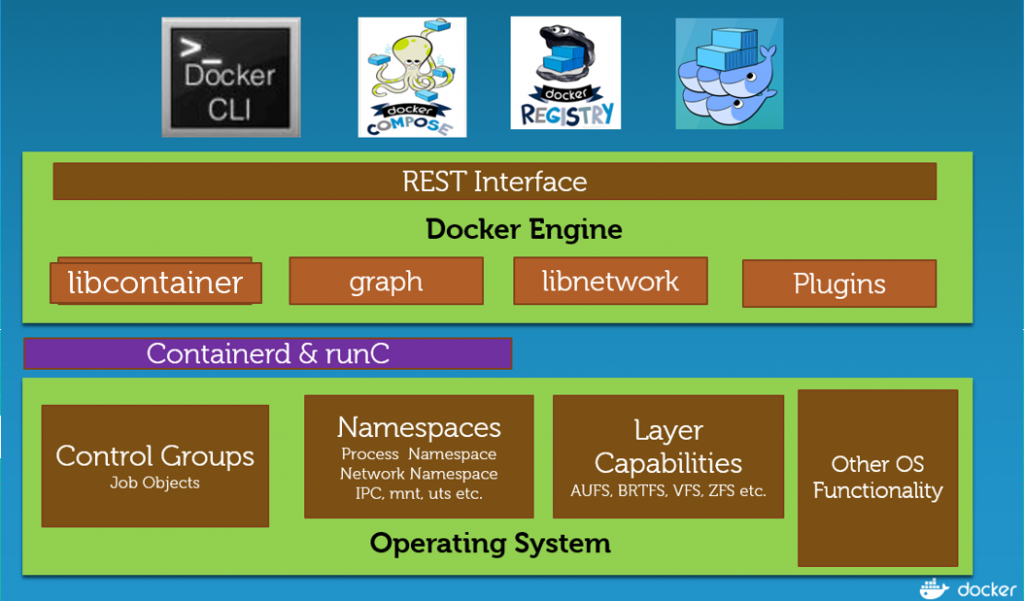
**Images**

An image is a read-only template with instructions for creating a Docker container.

**Containers**

A container is a runnable instance of an image.

* **Detailed Docker Architecture**



**containerd**

containerd spins up runc or other OCI compliant run time to run containers.

**runc**

runc is a CLI tool for spawning and running containers according to the OCI specification.

**Container format**

Docker Engine combines the namespaces, control groups, and UnionFS into a wrapper called a container format. The default container format is libcontainer.

**Namespaces**

Docker uses a technology called namespaces to provide the isolated workspace called the container. Docker Engine uses following namespaces pid,net,ipc,mnt,uts

**Control groups**

Docker Engine on Linux also relies on another technology called control groups (cgroups). A cgroup limits an application to a specific set of resources. Control groups allow Docker Engine to share available hardware resources to containers and optionally enforce limits and constraints.

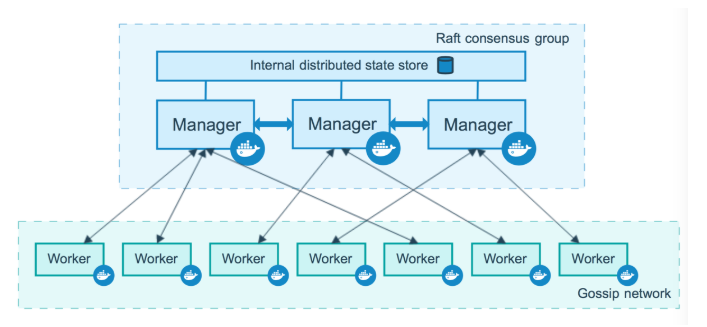
**Union file systems**

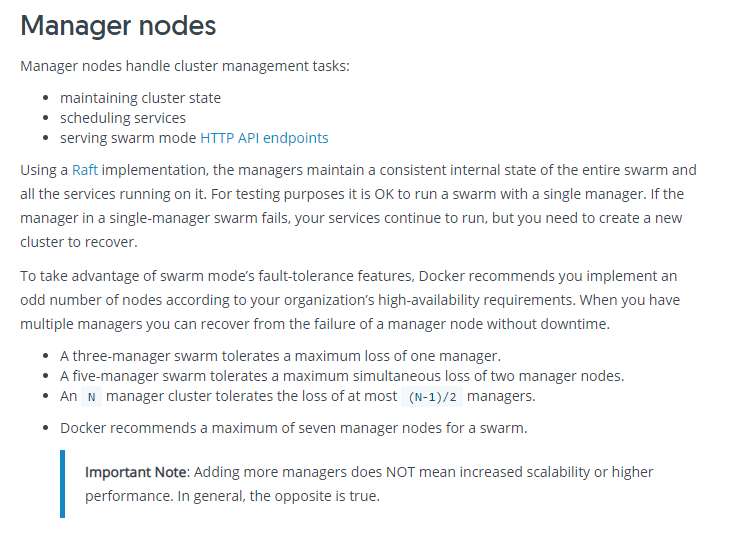
Union file systems, or UnionFS, are file systems that operate by creating layers, making them very lightweight and fast. Docker Engine uses UnionFS to provide the building blocks for containers. Docker Engine can use multiple UnionFS variants, including AUFS, btrfs, vfs, and DeviceMapper.

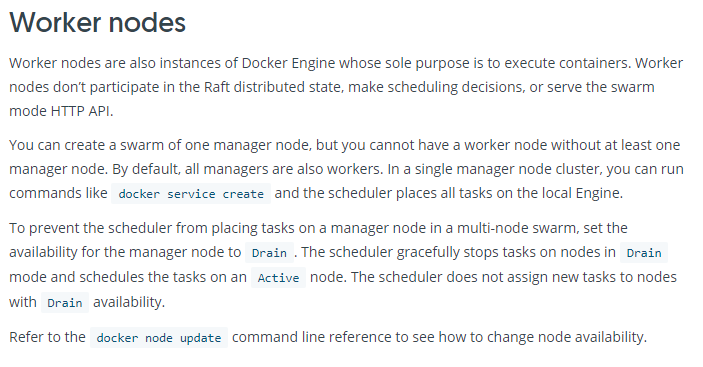
* **Docker Swarm**

Docker Swarm is a tool that allows you to deploy a cluster of Docker Hosts. It's a native clustering tool provided by Docker which provides high-availability and high-performance for your application by distributing it to all nodes inside the swarm cluster.

* **Docker Swarm Architecture**







* **Docker Compose**

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application’s services. Then, with a single command, you create and start all the services from your configuration.