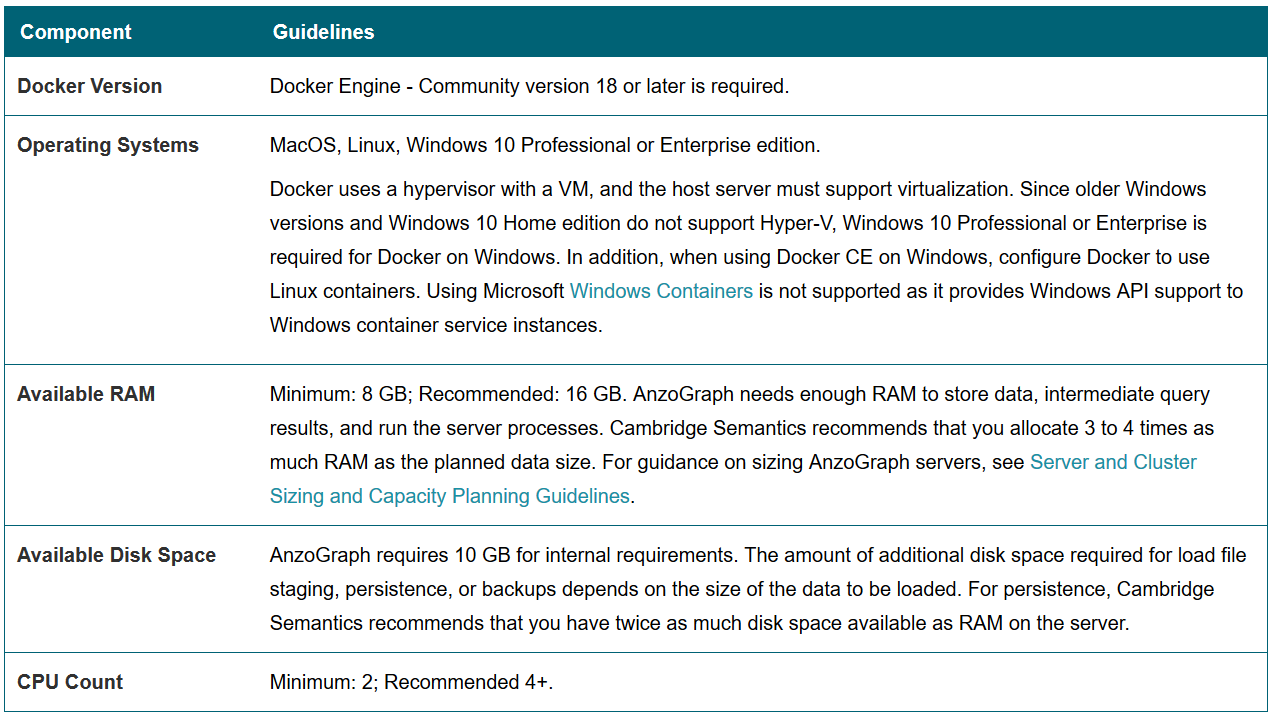
**Requirement:**



**Topics :**

1. Architecture & Documentation.

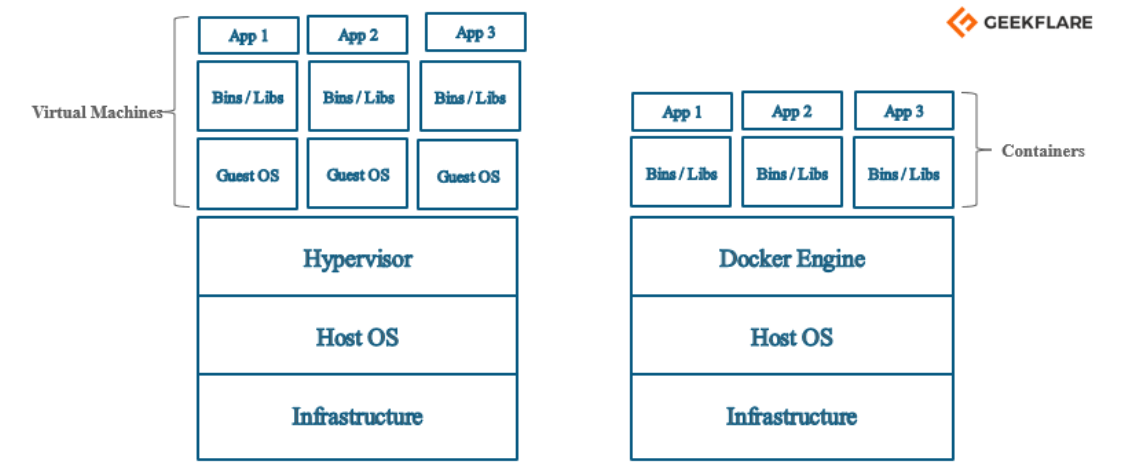
2. Installation

3. Docker All Commands Practical Session.

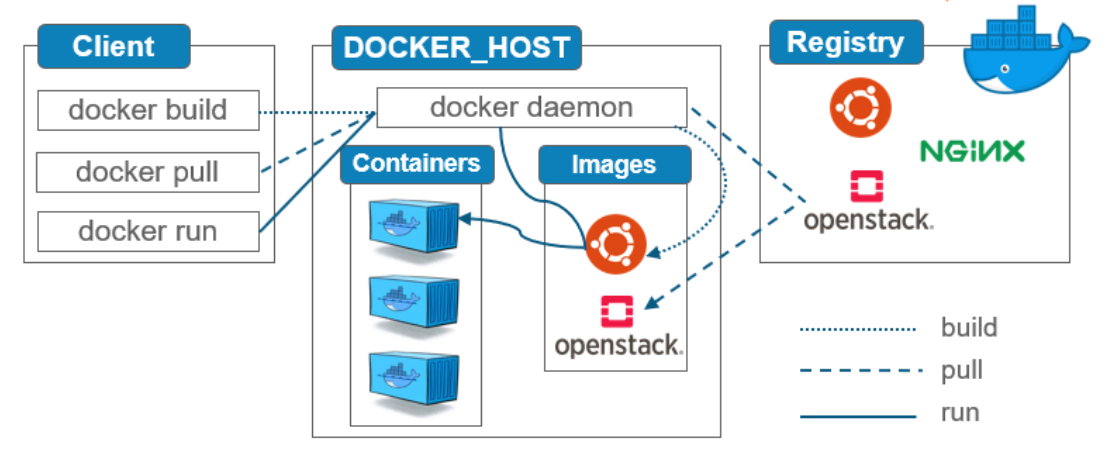
* Image & Container Creation with manual commands and Dockerfile.
* Volume Creation and related commands
* Container Network related commands

4. Docker Hub Registry Account Creation and their commands in CLI

Differences between Virtual Machine and Docker Containers:



**Docker Architecture:**



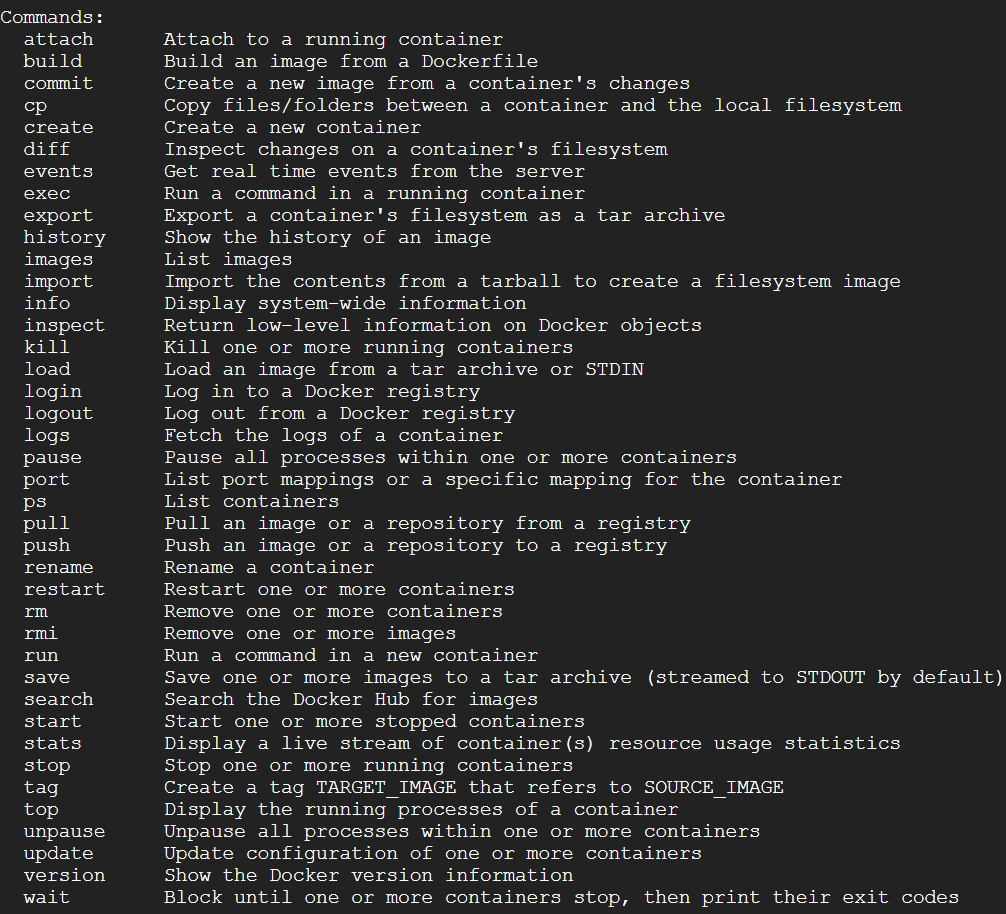
**Docker Installation links in both Windows and Linux:**

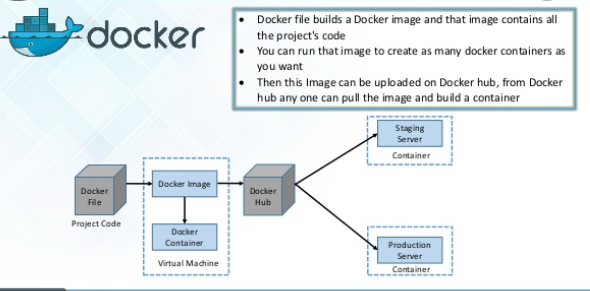
<https://docs.docker.com/docker-for-windows/install/>

<https://docs.docker.com/engine/install/linux-postinstall/>

<https://docs.docker.com/engine/install/centos/>

**Docker Practical Session Commands**





docker images

docker run -it centos

docker run -itd ubuntu

docker ps -a

docker start <container id>

docker stop <container id>

docker run --name <my container name> <image name>

docker search debian

registry/username/repo:tagname (official images doesn’t have user name)

docker.io/devopsadministrator2903/ubuntu

docker push devopsadministrator2903/ubuntu:tagname

docker volume ls

docker network ls

docker run -it --name <containername> -v <container Volume name in host>:<stored data container location> <imagename>

docker volume inspect <volume name in host>

docker cp <containername>:<file/folder inside the container> <location in host> (copying data from container to host)

docker cp <file/folder location in host> <containername>:<location to be copied in container> (copying data from host to container)

docker attach <container id>

docker exec <container id> <command to be executed>

--------------------

Creating Image from a running container / by using docker file( where we place docker instructions.

Creating our own custom images

docker run -it –name <containername> <imagename> /bin/bash

Install httpd in container and exit from container

docker commit <container id> <dockerhub username>/<imagename>:<tagname>

docker history <image id>

Now creating image through a docker file

[root@server ~]# pwd

/root

[root@server ~]# cat dockerfile

From ubuntu

Maintainer "Prasad"

RUN apt-get update

RUN apt-get install httpd -y

CMD /bin/echo "httpd installed successfully"

Save the file

Run the command

docker build -t <username/imagename>:<dockerfile> <path of dockerfile> /bin/bash

docker inspect <container id>

docker run -it -P tomcat (32768—port forwarding/exposing)

docker run -it -p 9999:8080 tomcat (binding ports)

docker run -itd -p 5000:5000 registry:2

curl -I <http://localhost:5000/v2>

docker tag devopadministrator2903/centos localhost:5000/centos:v1.0

docker push localhost:5000/centos:v1.0

curl -I <http://localhost:5000/v2/_catalog>

**FYR, If we want to install in windows**

Docker ToolBox

Docker Machine for running docker-machine commands

Docker engine for running the docker commands

Docker Compose for

Kitematic- Destktop GUI for Docker

Oracle Virtual Box

Shell preconfigured for docker command-line environment

**Commands:**

docker machine create --driver virtualbox node1

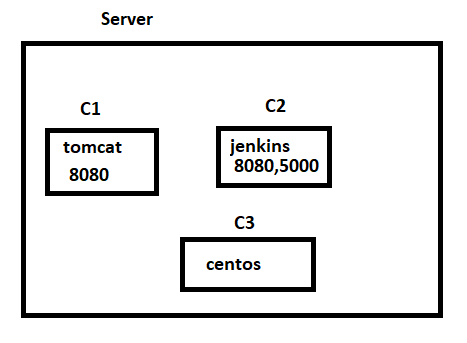
docker machine ls

eval “$(dockermachine env node1)”

**Docker Compose:**

**yum install python-pip**

**pip install docker-compose**



**create yaml file to create multiple containers:**

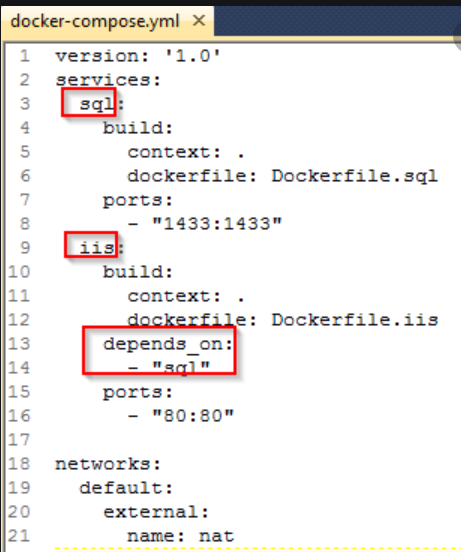
docker-compose up -d

docker-compose build

docker-compose scale SERVICE=3

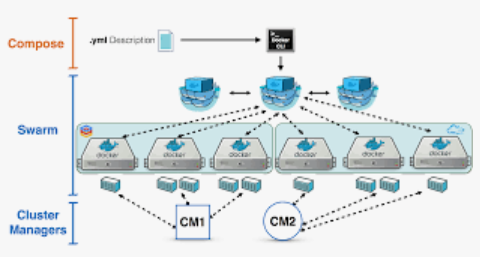
docker-compose up --no-recreate

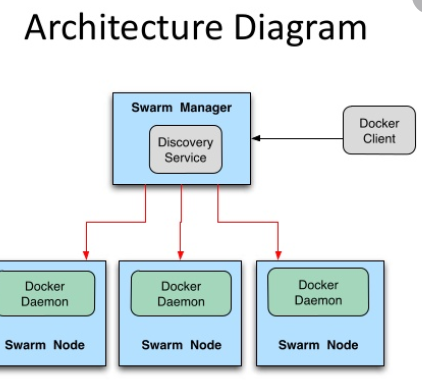
docker-compose stop



**Docker Swarm Architecture: Container Orchestration**

Multiple containers on docker hosts





**Advantages:**

HA

Load Balancing

Scaling

**Commands:**

Note: Docker must be installed in every node.

Docker swarm init –advertise-aadr <ipaddress> (It generates a token)

Docker node ls

We must copy token of manager to node in order to make the host as a docker worker nodes.

docker swarm join-token manager

docker swarm join-token worker

docker service create -p 8000:80 –name <service name> <image name>

docker service ls

docker service ps <service name>

docker node update –availability drain manager

docker service scale <servicename>=10

docker node update –availability active manager

docker service create –name <servicename> --network myoverlay -p 8000:80 <image name>