24/7/2017

* Docker automates the repetitive tasks of setting up and configuring development environments so that developers can focus on what matters: building great software.
* Docker – abstracted version of machine(hardware) whch runs ½ services
* Open source-underlying tech which docker uses.
* Does not have certification.
* Docker itself is not a container.
* All docker images are made public

Using containers, everything required to make a piece of software run is packaged into isolated containers. Unlike VMs, containers do not bundle a full operating system - only libraries and settings required to make the software work are needed. This makes for efficient, lightweight, self-contained systems and guarantees that software will always run the same, regardless of where it’s deployed.

* Platform which facilitates easy creation of containers.
* Initially started as open src
* Vm requires iso,docker requires image.
* Container is run-time env of image
* Competitors od docker- solaris,red
* But docker is quickest of all.
* Manage image and provides platform to run containers
* Namespaces and control groups-docker uses these to faciltitae containers.
* Uses tech of linux containers
* Built on linux containers, but Microsoft has planned to go open source, down the line windows containers will also be popular.
* Union capable file system[inbuilt file system]-mgmt of image-varies form one os to another which docker uses.[ex. Aufs is common platform],brfs,overlay
* Data shld be kept outside containers to maintain containers light.[Recommended]
* Namespaces- facilitates sharing of OS.
* Limiting of resources is important
* Control groups faciltitates limiting of resources
* Container communication is possible
* Never run container with root privilege
* Docker daemon comm shld happen with tls certificate
* Separate Docker networks
* Exsi – type1 hypervisor
* Docker containers – type 2 hypervisor
* All binaries are retieved from the base os

VM:

1.H/W LEVEL SYNC

2.HEAVY WEIGHT

3.FULLY ISOLATED

4.SLOW PROVISIONING

5.LIMITED PERFOMANCE

CONTAINER:

1.SYSTEM LEVEL

2.LIGHT

3.PROCESS LEVEL ISOL

4.REAL TIME ISOLATION

5.EASILY SCALABLE

BINARIES AND LIBRARIES ARE PLACED IN IMAGES.

COMMON BINARIES ARE SHARED.CONTAINER-SPECIFIC ARE KEPT INSIDE EACH .

MAKING CHANGES TO SHARED BINARIES:

THEY GET ISOLATED FURTHER

DOCKER LAYER TRACKES BINARY CHANGES.

WRITABLE LAYERS[CHANGES SPECIFIC TO CONTAINER] AS ADDED AS TOP LEVEL LAYERS.

FOR EACH CHANGE A NEW LAYER GETS ADDED.[WRITE-ONLY LAYER]

ARCH :

SERVER->HOSTOS->IMAGES[LIB/BINS]->CONTAINERS

DOCKER RUNS AS ADAEMON[SERVICE]

CLIENT-SERVER ARCH

2 THINGS:

1.PROCESSING RUNNING INSEIDE CONTAINER-TAKEN CARE BY NAMESPACE

2.DATA INSIDE CONTAINER-TAKEN CARE BY FILESYSTEM

CONTROL GROUPS:ENSURE EACH CONTAINER HAS ITS FAIR AMT OF SHARE.

2 WAYS OF CUSTOMISING IMAGE:

1.PROVIDED ISO, CREATE A CONTAINER,MAKE CHANGES,COMMIT IT BACK

2.CREATE A DOCKER FILE,

DAEMON-PRCOESS THAT MANAGES CONTAINERS

KERNEL->BASE IMAGE->

In windows,on installation of docker, it installs a hypervisor

wget -qO-https://get.docker.com/|sh

docker pull centos:6

docker history centos:6 - to find the no. of layers

docker info

docker ps – shows running containers

to see all containers - docker ps -a

container- start,stop,remove

If a container does not even have one process running, it will be stopped.

Container can be addressed by either imageid/name.[Automatically image id is generated]

To remove container- docker rm container\_name/id

Run – used to create new container from containers

Start- to start a container

Docker run -I -t –name containername image

To run and initialize container.

yum update

ps aux

exit – to come out of the container

ctrl p ctrl q – to come out of container without stopping it

docker attach container-name – to get into running container

docker exec container-name - if not in bin/bash

docker exec -i -t container-name /bin/bash

docker rm -f cont-name : removes the container even if its running.

Docker commit -m “first commit” cont-name

Gedit,touch,nano – create new files

Docker create –name cont-name image : only creates container but does not run it.

Docker run – provides additional capability of creating a process.

Larger the width of data, slower will be the operation of accessing the layers

Data is not added to the image but mounted.

Docker run -it –name vol1 -volumes-from vol2 centos:6

Mounting data does not add to the image. Mount point is necessary to access data volume from the committed volume