Big code divided into small small code.

docker is a concept of implemtation of microservicess.

What are microservices?

Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of services that are

* Highly maintainable and testable
* Loosely coupled
* Independently deployable
* Organized around business capabilities
* Owned by a small team
* Service maintained b
* Easily service monitoring

The microservice architecture enables the rapid, frequent and reliable delivery of large, complex applications. It also enables an organization to evolve its technology stack.

Application architecture patterns

* [Monolithic architecture](https://microservices.io/patterns/monolithic.html)
* [Microservice architecture](https://microservices.io/patterns/microservices.html)

Data management

* [Database per Service](https://microservices.io/patterns/data/database-per-service.html)
* [Shared database](https://microservices.io/patterns/data/shared-database.html)
* [Saga](https://microservices.io/patterns/data/saga.html)
* [API Composition](https://microservices.io/patterns/data/api-composition.html)
* [CQRS](https://microservices.io/patterns/data/cqrs.html)
* [Domain event](https://microservices.io/patterns/data/domain-event.html)
* [Event sourcing](https://microservices.io/patterns/data/event-sourcing.html)

Deployment patterns

* [Multiple service instances per host](https://microservices.io/patterns/deployment/multiple-services-per-host.html)
* [Service instance per host](https://microservices.io/patterns/deployment/single-service-per-host.html)
* [Service instance per VM](https://microservices.io/patterns/deployment/service-per-vm.html)
* [Service instance per Container](https://microservices.io/patterns/deployment/service-per-container.html)
* [Serverless deployment](https://microservices.io/patterns/deployment/serverless-deployment.html)
* [Service deployment platform](https://microservices.io/patterns/deployment/service-deployment-platform.html)

## Forces

* Services are written using a variety of languages, frameworks, and framework versions
* Each service consists of multiple service instances for throughput and availability
* Service must be independently deployable and scalable
* Service instances need to be isolated from one another
* You need to be able to quickly build and deploy a service
* You need to be able to constrain the resources (CPU and memory) consumed by a service
* You need to monitor the behavior of each service instance
* You want deployment to reliable
* You must deploy the application as cost-effectively as possible

Micro-servicess: means mincro management: micro architure.

Flipkat application services

Flipkart app maintaine micro arcture.

1. Login
2. Authication
3. Sales
4. Inventory
5. Payment
6. Dilavary Status
7. Report
8. Feedback

Thse application are titly coupled

When sales having more requests I want to incress sales copies for more servers but these applications are tightly coupled so not possible to scale up and scale down for one functnality.i want to keep entire application in servers only that is not good

Who getting more req that functnality only I want to scalling the servers.

When I want to entire application maintain scaling

I want to keep entire app in seervers servers want to good configuration only

So much cost is involved how to avoid go for micro services

Micro services means: spilt the every functnality by the developer only after the micro services evry functnality code is very light weight I want to run user registration I want 100 mb ram configuration only but privous I want to run one fuctnality not possible because of that code is tightly coopuled .

I want to done micro services done means loosly copled

Manage microservicess arctecture use docker.

Every functnality we can create as a image.

In docker maintaine micro services arciture using of image

Create image for user registration

Create image for authication

Create image for all funcnality

That image is light weight.

1. Now sales having high load creat for image for sales and I want to create more replicas(copies) for sales funcnality.
2. Now status having high load creat for image for status and I want to create more replicas for status funcnality.
3. Docker is tool to implement a micro services architecture . And that micro services architure maintaine purpose k8s/docker swarm all of compines are not using dockerswarm.
4. Doccker is only implementation of microservicess when one functnality load incress then more replicas want for that functnality in docker we will create manually

But in dockwer swarm/k8s it automatically replicas scalling and scaling down.

1. Maintaine means scalling .
2. What id diff b/w docker and k8s: in market multiple containers are avalible like dcker container and openshift containers but ds is only support for docker containers only
3. K8s: scalling the replicas distubuting load and high availability ,fault tolerance,disaster

This achive purpose we can use k8s.

K8s having autoscalling,load balancing,rolling updates,deployment statagies.

Your working for d-mart.realience mart,walmart

* 1. Order management,inventory(a complete list of items such as property),shipping delivaries,user registration,user autication,payment management,

Invoice(**request payment after work is completed**, and where there is an ongoing relationship with the customer.),purchase management

* 1. Data planes,inventory management,sim autication,sim status,order management.

1. How to check docker installed or not : docker – - version
2. Get docker.com

Curl --------

Install dockerget.sh file

Run that one that file using of sh command install docker engine.

Only for Ubuntu machine.

Docker info

Docker images we taken from docker public registry no need authication aws also maintain ecr

Docker registry contains docker images

How to check docker images or not

Docker images

Docker images light weight in mb s only.

Hub.docker.com—it is official docker reg.

In real time use private images

Docker pull image name: I will download from docker reg but I am not mention that docker registr info but it will pull from docker registry

How because docker is design not mention any thing it is taken images from docker hub only.

Image alwals having a tag

Ubuntu:latest

Ubuntu-image name and latest tag we not mention tag it taken latest tag.

Docker version but docker point it is a tag.

Nginx:1.23.3 tag=1.23.3

Image converted into container

Hoe to check contaire runs docker ps

All containes docker ps –a: it display shown all containes

1. How to run docker containe

Docker run imagename: but when using this commad what process runs in containe it will shown only we canot do anything

1. What is diff between ami/instance:

Ami: it contains os. Using of ami we can launch ec2.

Docker image: it contains os only using of docker image we can create container

What is diff between ami and image .

Ubuntu Ami –using of ami I created instance and on that instance I want to run a java app so I install tomcat in that machine.

Here on instance why tomcat and java app installed on that machine.

That Ubuntu ami having libiries and configuration files to supports tomcat,java app

Ubuntu ami contains libiries and conf files to install and run the all softwares .

That ami is not a light weight.

Docker image containes os but it is light weight os(bare os) and It containes only liberies of run the your application.

This is not containes all liberies of all softwares what software you want to install that liberies only it having.

Docker image containes bare os it is not support curl,wget all those thing s.

It containes simple command runs liberies only.

DOCKER IMAGE : DOCKER IMAGE CONTAINS BARE OS(required to run baae os)0 AND REQUIRED LIBERIES TO RUN YOUR APPLICATION.applicating might be tomcat or nginx.

Docker image=bare os + liberies

What is use of image : we can easy to ship it

Ami is the heigt weight it runs to want high configuration machine

Docker image no need high configuration machine.

1. What is diff between the vm and container

I have machine in that I need to install hypervizer software

Why we need install hypervizer software : htper vizer our hardware divided into diff independent machines . we can install multiple os.on individual machines.

Parts.

We are not having hypervizer software in bare metal we want to install os that os occupy all hardware resourcess bare metal.

Hypervizer: when we install hypervizer in bare metal we can divided into multiple virtual machines on that virtual machines we can install os.

**ubuntulinux**

**Vm1 redhat linux**

**4gb ram 2 cpu 30 gb storage Vm2**

**1gb ram 1 cpu 30 gb storage centos**

**Vm3**

**4gb ram 2 cpu 30 gb storage**

**======install hypervizer software**

**16gb ram 4 cpu 100 gb storage**

**Own vm having own os and own vm containes own ip address.**

**For example: application runs on vm1.app2 runs vm2,app3 runs vm3.**

**App1 having high load that configuration not enough**

**Vm2 not having any load**

**It is possible to swap vm2 ram to vm1 : it is not possible to swap os resources(ram,cpu,storage) in virtulization**

**It is creation of vm1 only we can allocate that is fix we canot swap.**

**C1 c2 c3 c4**

**Docker engine**

**os**

**Bare metal**

**When bare metal we can install os on that os we can install docker software from that docker software we can create containers . in containers want os resources it will taken from that machine it will used from comman area.**

**Container having own ip address.**

**For example: application1 runs on c1.app2 runs c2,app3 runs c3.**

**App1 having high load that configuration not enough.**

**C1 taken os resources from comman area and it will used.**

**We are given any os resources at time of container creation.**

**When container need os resources os resources having comman area container take it and use.**

**We are install only one os on bare metal and container having base os.**

**In virtualization each and every vm wants own os and we can not swap os resources.**

**When ever wants os resources vm’s are looking vms os resources**

**Containers wants os resources it will shown bare metal.**

**Windows**

**Vm1 redhat linux**

**4gb ram 2 cpu 30 gb storage Vm2**

**1gb ram 1 cpu 30 gb storage centos**

**Vm3**

**4gb ram 2 cpu 30 gb storage**

**======install hypervizer software------------------it is possible**

**16gb ram 4 cpu 100 gb storage**

**C1(windows) c2(linux) c3(ubuntu) c4(centos)**

**Docker engine**

**Os -----it is not possible**

**Bare metal**

**think on bare metal diff vm vs docker**

**12) docker image**

**Multiple compines create a docker images their requrmnt**

**Oracle make oracle image : it contains minimul os+liberies of to run to oracle software**

**Nginx make nginx image: it contains minimul os + liberies of to run nginx software**

**Docker logo: means we can containers or image easily ship one machine to another machine**

**What ever thing your application we will putting into the image .**

**Image can shipped into one machine to another machine.**

**13) instalization of docker:**

**get docker.com from website we can install docker that is not official website.**

**That all commands keep into one script we can run that script it will automatically installed on any linux flavour.**

**This script is meant for quick & easy install via:**

**# $ curl -fsSL https://get.docker.com -o get-docker.sh**

**# $ sh get-docker.sh**

**To check docker is avalible or not**

**Docker version**

**Docker –version**

**Docker info**

**Mainy we are using version 19 or 20.**

**Docker container : run a container I need docker image.**

**Image are stored in docker registries in repository.**

**Docker maintaine docker registry all images docker maintain in that registry**

**Ecr: fully managed images.**

**Docker images: it will shown docker images**

**Docker pull imagename: it will pull the image from docker hub**

**Docker software defined by default it will downloaded images from docker hub only.**

**Images having multiple versions. In docker understand tags**

**Tags=version.**

**Docker ps: it will display running containers**

**Docker ps –a: it will show all containers.**

**Docker run imagename:nginx container running but it is running in attached mode.**

**Attached mode means it will show like proceser we cannot run any command in that screen.**

**Docker run –d nginx : I run a container detached mode it will shown container id**

**Detached mode means : container runs on backside.**

**Container process runs backside.**

**Each and every container have container id**

**Nginx container running inside nginx app1 is running**

**Nginx image contains:bare os +lib for running nginx server**

**Inside machine container inside container app running.**

**I want acess:63.35.26.25:80------not accessing**

**I am giving send req to host os in that host ip:80 no app is running**

**But my application runs on container port**

**So**

**How to send req to container port**

**Host os means where docker container runs that is host os.**

**Host os containes 0-65535 ports.**

**Instance(ip)**

**Cont1(ip)-----appl—80.**

**Cont1(ip)-----app2—90**

**Cont1(ip)-----app3--100**

**User send req to host os port 80 but ther 80 not app running app1 running in container port 80.**

**How I send that host port 80 to container port.**

**Port mapping: -p 80(host os port):80(container port)**

**User send req allways to host os it will take send req to cont1 cont send req to container port**

**80.on 80 nginx running.nginx take req and excuting send a respose back.**

**When ever running the container I need to map container port with host port.**

**-p 80(host port):90(container port or container app port)**

**What ever end user send req to host port I am forwarding to container port.**

**App is running inside container you want to acess from out-side we need to do the port mapping.**

**Docker run –d –p 80:90 nginx**

**This container exposed to outside**

**Docker stop cid**

**Docker exec –it(interactive terminal) nginx bash**

**-it means: container terminal attached to the your console.**

**Docker registry**

* 1. **Docker stop cid**
  2. **Docker ps**
  3. **Docker ps –a**
  4. **Docker images**
  5. **Docker info**
  6. **Docker --version**
  7. **Docker pull imagename**
  8. **Docker rm cid1 cid2**
  9. **Docker run –d –p 80:90 image name**
  10. **Docker exec –it(interactive termina) image bash**
  11. **Docker rmi**
  12. **Docker rmi imagename—delete images**

**Docker**

**-d ---container runs on deteached mode**

**-p-----using port number mapping**

**13) docker –name --hostname**

**14) docker build –t**

**15) docker volume**

**16) docker purne**

**17) docker logs**

**Tty –termina**

**Terminal also treaded as file in linux**

CONTAINER ID 5027e7bc74b6 -cid

IMAGE nginx

COMMAND "/docker-entrypoint.…"( –very important)

Here command dock-entrypoint processer excuting first instruction.

STATUS **Up 27 minutes –up means it s running stage**

PORTS **0.0.0.0:80->80**/tcp, **:::80->80/**

Any one give req to Host port – it will send to 80 con port

Docker run –it imagename----

**Docker run –it imagename bash**

**When ever we are open server or machine process will run**

**Otherwise it will not open**

pts/1 00:00:00 bash

**process start do some work and die**

**linux point of any command is process only that process have one pid**

**that process do some activity**

**when ever container start process will start.**

**Cmd will excute . which process will start that information having in that command**

**Command having piece of code.**

**Ps –u**

**-it : means our assign to our console to docker container.**

**Image—bare os + lib req to run our app.**

**When ever go to container we can run processer . like nginx,tomcat, by default nginx process start command already in nginx image using docker file we are adding extra informantion only we are not create to base.**

**Conainer**

**Docker run –p 80:90**

**Docker exec –it**

**Docker run –d**

**Docker pull**

**Docker images**

**Docker ps –a**

**Docker info**

**Docker –version**

**Docker info docker image**

**Docker info cid**

**I given cmd instruction that instruction it will excute first when container will run(created)**

**Cmd will over ridden**

**When creation of container I am not giving cmd instruction(argument) by default cmd instruction excute in image what ever image having cmd instructions .**

**Some excuting cmd’s will give in cmd instruction**

**Cmd 🡪 ls in cmd instruction we give ls ,ls cmd it will excute**

**Cmd : default/overwritten.**

**Entrypoint: there is no way to over written. But we can pass additional activities**

**Entry point instructions excute by default when ever container start first cmd excute entrypoint instruction when we pass aditinoal values it will appended to the entrypoint.**

**Entry point instruction given in image it will excute .**

**When container will run which process(cmd) will run firsrt**

**Imag1**

**CMD [ls]**

**Docker run –d image1**

**-when container will start in side container cmd instruction will excute first**

**Ls---will excute first**

**Docker run –d image1 cat /etc/passwd----**

**Ls overwritten cat /etc/passwd so when container run /etc/passwd**

**Will run.**

**Image2:**

**EntryPoint [ls]**

**Docker run –d ihage2 ===🡺 ls will execute**

**Docker run –d image2 /etc =====ls /etc will execute**

**Docker run –d image2 sheshi=====ls sheshi it will will execute**

**Image 3 : containing two instruction cmd and entrypoint**

**Entrypoint [head]**

**Cmd [-n 10 /etc/passwd]**

**Cmd instruction append to entrypoint**

**When cmd and entrypoint both in docker image cmd instruction append to entypoint instruction**

**Docker entrypoint.sh;ls -l**

**Docker run –d image3 -----🡪 head –n 10**

**When ever container created that command will excuted**

**Docker run –d image3 –n 20 /etc/showd**

**Cmd over written to –n 20 /etc/showd and this cmd append to entrypoint**

**Head –n 20 /etc/showd**

**We are having multiple entrypoint and cmd s then only one entry point will taken 1st instruction**

**We are image is new to us first we can check image .**

**Bash is allways req terminal**

**Docker inspect imagename.**

Cmd": [

"bash”

],

**Docker run –d - –name c1 ubuntu**

**Bash allways want terminal we canot assign to terminal so that process will be die**

**Container will be exited state.**

**Cmd—“bash” –it will having bash command cmd instruction in image.**

**Bash command is excuted but terminal we not attaching to bash command.**

**When ever we assin to terminal to bash it will give**

**we need to give terminal to bash command then only it will give container terminal attaching to our console**

**bash: this cmd will do container want to attach to terminal so bash want terminal that terminal our responsibility assign to bash.**

**Docker host port**

**Docker run –it - - name c2 –p 80:80 Ubuntu bash**

**It will goto container.**

**Exit**

**When exit container existed stage**

**Docker ps –aq –it will give all cid’s**

**Docker rm $( Docker ps –aq)**

**Docker run –d –name c1 ubuntu bash**

**Bash excuting inside contaoiner bash want terminal we are not attaching terminal so bash is one process to that os .**

**Bash command not having terminal so it is die.**

**So out container exicted state.**

docker run -it --name c3 ubuntu---it will got to container inside.

**Docker run - –name c4 ubuntu ls –l**

**Ls –l output come.**

**Docker run - –name c4 ubuntu /etc/passwd**

**Excuted error came container is exited state**

**In real-time docker container we run but container goes to exited state means not running**

**We are facing with some issues in docker image we have to tell.**

**In existing thing any wrong we have to tell.**

**Docker run –d nginximage ---------🡪/docker-entrypoint.sh nginx –g demon.off**

**When container created(run) from image that time cmd instructions will excute**

**Not every time.**

**Docker exec –excuting some commands inside running conatiners**

**Bash command excute for output shown putpose bash command wants terminal**

**Bash and sh commands shown output they want terminal**

**Except 2 cmds all cmnd’s not require terminal it will shown in server.**

**Docker exec cid printenv**

**Server have its own ip address like container have its own ip address**

**Interview q:how to find container ip address.**

**Docker inspect dockerid**

root@ip-10-0-0-19:~# docker inspect c67a84fc507a | grep IPAddress

"SecondaryIPAddresses": null,

"IPAddress": "172.17.0.2",

"IPAddress": "172.17.0.2",

**WHEN EVER WE STOP AND START THE SERVER DATA WILL BE AVALIABLE**

**BUT**

**CONTAINER STOP AND START DATA WILL be there**

**When container die data will be gone**

**Container are fmrun**

**When container die we don’t dnow it is die inside container data gone**

**Now we are creating using of same image data will not come**

**How to maintaine docker data persistant.**

**Instance---🡪 we can attach disk(drive).**

**Container having it’s own directory structure.**

**instance having it’s own directory structure.**

**I want to map con dir to the docker host dir.**

**Docker container disk I mounted to docker host dir.**

**Host dir will mapped to the docker volume.**

**Interview q: what is default docker volume loc**

**/var/lib/docker/volums/**

**What ever**

**Docker inspect**

**Docker volume:**

docker run -d -v /app nginx : /app is docker volume it is attaching to the host dir

what ever we changes in docker volume(/app) it changes in host dir also

vice versa.

That docker volume dir attaching host os dir location /var/lib/docker/volums/

That dir maintaine by docker only default location

**We want to change loc it is possible.**

**Docker run –d –v /root/d1:/sheshi Ubuntu**

**/root/d1—this dir I want to mounted to /sheshi-dir when creation of docker container only.**

**In root(/) sheshi not ther it will create.**

**Docker inspect cid—important**

**Ther is way to docker data is persistant.**

**Container data we make persistence. Using docker volume.**

**Con having bareos (that process run want that code will be thre)**

**Docker pull**

**Docker run------create a new container**

**Docker exec ---login into the running container**

**Docker stop**

**Docker ps**

**Docker ps –aq**

**Docker inspect---/image/container**

**Docker rmi**

**Docker rm---🡪docker rm –f**

**Docker volume ls**

**Docker images –aq**

**Docker rmi $(docker image -aq) –delete all images**

**Docker rm $(docker ps -aq)—delete docker all containers**

**Docker con are when server in stoped stage con also exited condition.**

**===============================================================**

**How to create custome images;**

**FROM : means which base image is you taken**

**WORKDIR: WE LOGIN INTO THE CONTAINER OUR COMMANDS EXCUTES IN WHAT EVER WE SET WORKING DIR**

* 1. **Image build: some instruction excuted when creation of image building(CREATION)**
  2. **Based on image running container: some instruction are excuted at time of container creation**

**RUN :RUN IS EXCUTED WHEN IMAGE CREATION. IWANT TO EXCUTE SOME COMMANDS**

**Mainy using cmd software packages installing .paches installing**

**COPY: YOU WANT TO COPY FILES FROM LOCAL MACHINE TO IMAGE**

**WE ARE USING COPY COMMAND.**

**THIS COMMAND IS EXCUTED WHEN IMAGE CREATION.**

**ADD: YOU WANT TO COPY FILES FROM REMOTE(EXTERNAL) MACHINE TO IMAGE**

**WE ARE USING ADD COMMAND.**

**THIS COMMAND IS EXCUTED WHEN IMAGE CREATION.**

**CMD/ENTRYPOINT: THESE BOTH ARE EXCUTED WHILE TIME OF CONATINER CREATION.**

**ARG:**

**ENV:**

**MAINTAINER**

**EXPOSE : TIME OF CONTAINER CREATION.**

**WHAT DOCKERFILE CONTAINE**

**WHAT IS DIFF B/W CMD/ENTRYPOINT**

**WHAT IS DIFF BETWEEN COPY/ADD**

**WHAT IS DIFF BETWEEN ARG/ENV**

**FROM STATEMENT IS MANDATORY ALL ARE OPTIONAL.**

**Container running stage image is not possible to delete**

**Docker rmi –f $(docker ps -aq)**

**All containers are stoped then images deleted.**

**I want to create custom images using our docker file.**

**Dockerfile : means we are taken from image then that image we are adding additional information.**

**Dockerfile:**

FROM tomcat

WorkDir /app

Docker build –t imag1:latest .

======================================

Dockerfile1

FROM tomcat

WorkDir /app

RUN apt-get update

RUN apt install net-tools

Expose 90

Docker build –t httpd:v1 –f Dockerfile1 .

FROM tomcat

WorkDir /app

RUN apt-get update

RUN apt install net-tools

Expose 90

COPY ./index.html /usr/share/nginx/html

Docker build –t httpd:v1 –f Dockerfile1 .

Docker run –d –p 80:80 –name c1 httpd:v1

=========================================================

1.sh file

Pwd

Echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*sheshi\*\*\*\*\*\*\*\*\*\*\*

Hostname

Ls –l

Cp /etc/passwd /myusers

Dockerfile2

FROM Ubuntu

WorkDir /myapp

COPY 1.sh .

CMD[“sh”,”1.sh”]

Docker build –t image:v1 –f Dockerfile2 .

Docker run image:v1

1.sh excute in docker container that out will shown ij console

===============================================

Using docker file creating custom images

Docker to implement microsevicess

Docker image containe bare os + app related lib

===============================================

Docker file contain:

From

Maintainer

Workdir

Run

Copy

Add

Cmd

Entrypoint

Expose

Arg

Env

==================================================

Docker run –d –name c1 –hostname c1 –p 80:80 –v /root:/app

Docker exec –it cid bash

Docker image contains bare os and having lib+conf for required to ru our application

===========================

Container contains application

How to acess container app using at time of con run we mention –p 80:60

How to excute a command running container

I want to excute cmd ls how?

Docker exec cid ls

When we to go we want terminal we providing to bash at time of going

Docker exec –it cid bash

====================================

How to docker data persistant using docker voume –v

By devalut docker volum loc

/var/lib/dock/volims/

We can create our own loc

===========================================

How to create docker image

Using from

Run

Copy

Add

Cmd

Entrypoint

Env

Run

Expose

Arg

maintainer

====================================================

What is diff b/w run and cmd

Run: it excute at the time image creation for using software packages installing system updates etc ..

Cmd: it excuteing at the time of container creation cmd instruction will excute first instruction excuting in container.

======================

What is diff b/w cmd and entrypoint

Cmd: it will excute when time of container creation cmd instruction can be overwritten.

At time of container creation we give instruction that instruction will overwritten cmd instruction

Entrypoint instruction excute at time of con creation that is first cmd excuting in con entrypoint can not be overwritten if we are trying to over ritten that instruction will append to entrypoint.

When in docker file cmd and entry point will be there cmd instruction appen to entry point instruction

Cmd and entrypoints multiple is there lasts one is excute.

Append example

Dockerfile

EntryPoint ls

Docker run –d imagename /etc

ls /etc --/etc append to entrypoint instruction like that

================================================

-it—interactive terminal

--name—con name

--hostname—con name inside con

-v hostdir:condir

-p host port: con port(app port)

Docker build -t –using of this cmd creating docker image using docker file

Repository called as image

My docker image in docker reg

Syntax:

Docker.io(docker registry)/sheshi(user crediantls)/Ubuntu(repository or image):v1(version or tag)

=========================================================

1. Docker logs cid----it will show the what inside docker docker cmd’s execution on consoal
2. Docker run command : it excute at time of image creation mainly it will use activities done using run command

That iamge contains activity when con creation using that image con getting that activity

Run mainly use of software packages excuting and usercreation patches installing updating containers .

1. Docker purne
2. When we are using nginx con automatically run nginx processer because inside cmd instruction runs service nginx start so we are trying to overwriten cmd instruction in docker file or docker ecec nginx start over written so that con excited stage
3. When copy using copy cmd we want to menstion source dest and destionation also.
4. Docker cp sourcdest dest : using of local file copying to running con at the time of any time not in docker file

==============================================================

Docker tasks:

1)launch ec2-instance

2) install docker

Using get.docker.com

1) Check docker version

Docker –version

Docker info

====================================================

1. Create nginx container
2. Check nginx con logs
3. Stop and delete the step 1 con
4. Create new nginx con and acess nginx app from 8080(host)
5. Check the ipadress of step 4 container—docker inspect conid
6. Create new nginx con and keep name c1 and acess nginx app from 9090(host)
7. Download usr reg html keep file name index.html
8. Copy the file index.html to c1 container to application path
9. Acess the app from browser

=======================================================

1. Create a dockerfile
   * 1. Download sample user html file name it as index.html
     2. Use nginx image
     3. Set workdir /usr/share/nginx/html
     4. Copy index.html file

2) build docker image name it as webimage

Docker build –t webimage .

3)create container based on step2 image,acessapplication port 80(hp)

FROM nginx

WorkDir /usr/share/nginx/html

Copy index.html /usr/share/nginx/html

Docker build –t webimage:v1 –f .

================================================

1. Create a docker file
2. Use nginx image
3. Create sheshi user
4. Build docker image,name it as webimage:v1
5. Create con based on step2 iamge,acess app on port 80

From nginx

Run useradd -m sheshi

===============================================

1. Create docker file
2. Create smaple shell script with belo content name 1.sh

Echo welcome to docker world

Echomy name is $USER

Echo \*\*\*\*\*\*\*\*\*\*bye\*\*\*\*\*\*\*\*\*\*\*\*\*

1. User Ubuntu image with 16 version
2. 1.shsceript need to excute at the time of con creation
3. Give ecxute permission on 1.sh
4. Create con
5. Check container logs

===========================

Create docker file

1. Use Ubuntu image
2. Excute “cat /etc/passwd” at the time of container creation

From Ubuntu:16.04

Cmd [“cat”,”/etc/passwd”]

=======================================================

Create a docker file

1. Take Ubuntu image
2. Define env variable “name” ita value is “sheshivardhan
3. **Print step2 env value when container is creted**

**FROM ubunt**

**ENV name=”sheshivardhan”**

**Cmd [“echo”, “$name”]**

**DOCKER RUN –IT –E COURCE=TEST MYIMAGE BASH**

**WE CAN OVERRIDDEN ENV VARIABLE USING -E**

**b)create a docker file**

**1) create 1.sh file with below cmds**

**Pwd**

**Id**

**Echo $USER**

**TOUCH /APP/F1**

**SLEEP 300**

**B) TAKE UBUNTU IMAGE**

**CONTA WORK DIR MUST BE “/APP”**

**CREATE USER SHESHI**

1. **TAKE SAMPLE JAVA HELLOWORLD PROGRAM**

**COMPILE THAT PROGRAM USING JAVA CONTAINER**

**STORE THE “CLASS” INTO UR LOCAL MACHINE**

**===========================================================**

**Docker run –it –name c1 Ubuntu bash**

**It will go to Ubuntu container then start process exit**

**Now Ubuntu inage running stage**

**In Ubuntu cmd instruction bash excuted when container created then cmd excuted**

**Then excute cmd in Ubuntu image using exec**

**====================================================**