MK

MAHENDRAKUMAR V

IT | CLOUD ENGINEER

Author: MahendraKumar V
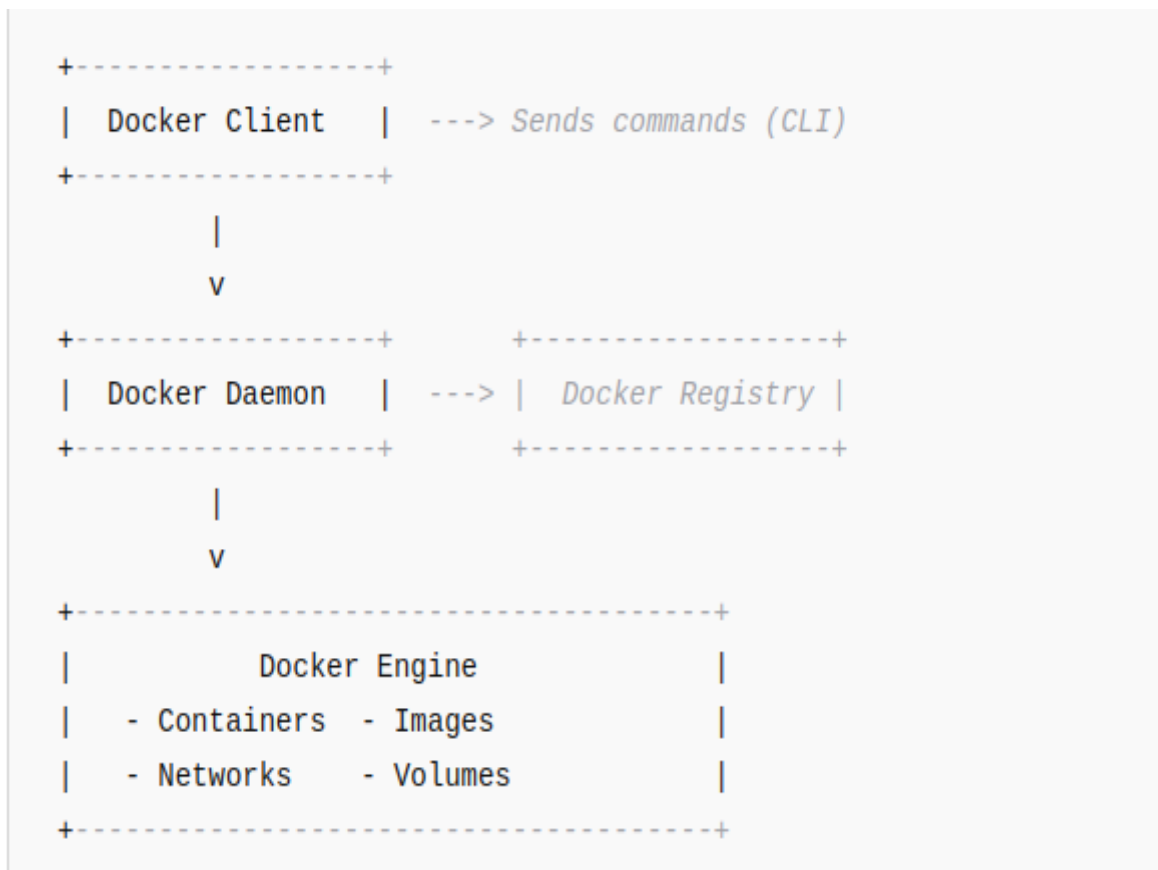**LinkedIn: https://www.linkedin.com/in/mahendra-kumar-v/**
Role: Cloud Engineer

1. Explain Docker Architecture?

Docker is a client – server architechture.

That allow user to users to build, package and run application.

Docker consist several architecture.

```
+--------------------+
|  Docker Client    |  ---> Sends commands (CLI)
+--------------------+
          |
          v
+--------------------+         +--------------------+
|  Docker Daemon    |  --->  |  Docker Registry  |
+--------------------+         +--------------------+
          |
          v
+------------------------------------------+
|            Docker Engine                 |
|   - Containers   - Images                |
|   - Networks     - Volumes               |
+------------------------------------------+
```

Docker client – Docker cli (User intract with Docker)

Docker Daemon – Listern API request and manage containers, images, Volumes and networks.

Docker engin – Runtime that allow container to run.

Docker image – Read only template used to create a docker container.

Docker container – Lightweight, portable and isolation instance of running application

Docker registry – Centralized repo for storing docker images.

Docker Networks – Create a communication between container Securely.

Docker Volumes – User persistant storage across containers restart.

2. What is FROM Statement in Dockerfile?

From instructions used which images to download and start from.

From instruction is a first command of docker file.

Its a manditory instruction, and its define the base image.

Syntax:

FROM <Base Image>:<tag>

Base images: Ubuntu, Alpine, NodeJs, Python...etc

Tag : Latest, 20.04..etc

Using official Base image:

FROM Ubuntu:22.04

Using lightweight images

FROM Alpine:Latest

FROM instrctions helps to optimize image size and minimize dependacies.

```
mahendrakumar-v@thinkpad-e14-gen-3:~/Study/Microservie_App/Microservice_Project/ecommerce-ui$ cat Dockerfile | grep -i "FROM"
FROM node:14
mahendrakumar-v@thinkpad-e14-gen-3:~/Study/Microservie_App/Microservice_Project/ecommerce-ui$ cat Dockerfile | head -n 10
# Use the official Node.js image as the base image
FROM node:14

# Set the working directory inside the container
WORKDIR /app

# Copy the server code to the working directory
COPY server /app/server

# Change to the server directory
mahendrakumar-v@thinkpad-e14-gen-3:~/Study/Microservie_App/Microservice_Project/ecommerce-ui$
```

3. Diff btw COPY and ADD in Dockerfile?

ADD and COPY instruction use cases are similarly

COPY : Copy files and Directorys from the local hostmachine in to inside the docker image

There is no other functionalities.

ADD: Using add instructions can extract compresses files and copy files from a remote location via URL.

Local compresed files are copy in to remote files using ADD instruction.

But in RUN Instrcutions supports both curl and wget is safe and more efficiency.

RUN curl http://source.file/package.file.tar.gz \

 | tar -xjC /tmp/ package.file.tar.gz \

  && make -C /tmp/ package.file.tar.gz

```
mahendrakumar-v@thinkpad-e14-gen-3:~/Study/Microservie_App/Microservice_Project/ecommerce-ui$ cat Dockerfile | grep -i "COPY"
# Copy the server code to the working directory
COPY server /app/server
# Copy the client code to the client directory
COPY client /app/client
mahendrakumar-v@thinkpad-e14-gen-3:~/Study/Microservie_App/Microservice_Project/ecommerce-ui$ cat Dockerfile | grep -i "RUN"
RUN npm install
RUN npm install
RUN npm run build
# Run the Node.js server when the container starts
mahendrakumar-v@thinkpad-e14-gen-3:~/Study/Microservie_App/Microservice_Project/ecommerce-ui$
```

4. Diff btw CMD & ENTRYPOINT in Dockerfile?

Both CMD and ENTRYPOINT define the default command that run those command when a container starts.

CMD : Provide a default command can be overriden when running a container.

ENTRYPOINT : is a fixed behavioue, We can't override, we can pass additional arguments.

mahendrakumar-v@thinkpad-e14-gen-3:~/Study/Microservie_App/Microservice_Project/ecommerce-ui$ cat Dockerfile | grep -i "CMD"

CMD ["node", "server.js"]

Here ENTRYPOINT is a main process.

If the new command passed ENTRYPOINT can't be overridden.

5. Diff btw RUN & CMD & ENTRYPOINT in Dockerfile?

RUN – Execute command in new layer. Creating a new image.Commonly used for installing software package

CMD – Is a default comand. To run a command when container starts.

ENTRYPOINT – is a fixed executable, Configure container to run as an executable.

6. Diff btw ENTRYPOINT & ARGS?

ENTRYPOINT – is a fixed executable, Configure container to run as an executable, Its a fixed command that always run when the container starts.

ARG : Define build time variable, That can be pass when building the image.

Ex: During the docker build.

FROM ubuntu:latest

ARG MY_VERSION=1.0

RUN echo "Application Version: $MY_VERSION"

7. How will you expose your container?

When docker container expose in outside using port, You can expose port in different method.

Ex: if you have web application running inside a container with the port 8080, You can epose it to the host system by running the container.

Running container you can use -p or –publish flag followed by the format

docker run -d –name <Docker container name> -p <Hostport>:<Network Port> <Image name>

root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker run -d --name Expose2 -p

```
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker run -d --name Expose2 -p 4001:4001 order-management
8fdf4a7bf1c37cd2f74d12bc480096b645273eff471bd9158187e87cec43db7a
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED          STATUS                    PORTS                                                                 NAMES
8fdf4a7bf1c3   order-management       "/usr/local/bin/mvn-…"   5 seconds ago    Up 4 seconds              0.0.0.0:4001->4001/tcp, :::4001->4001/tcp, 9090/tcp                   Expose2
20195c665396   ecommerce-ui           "docker-entrypoint.s…"   6 weeks ago      Up 14 hours               0.0.0.0:4000->4000/tcp, :::4000->4000/tcp                             ecommerce-ui2
def35cf22b4e   product-inventory      "python3 inventory_a…"   6 weeks ago      Up 14 hours               0.0.0.0:3002->3002/tcp, :::3002->3002/tcp                             product-inventory2
ccc840f8d96d   product-catalog        "docker-entrypoint.s…"   6 weeks ago      Up 14 hours               0.0.0.0:3001->3001/tcp, :::3001->3001/tcp                             product-catalog2
b33cd333983e   contact-support-team   "python3 server.py"      6 weeks ago      Up 14 hours               0.0.0.0:8000->8000/tcp, :::8000->8000/tcp                             contact-support-team2
4a6f60b2d075   order-management       "/usr/local/bin/mvn-…"   6 weeks ago      Up 14 hours               0.0.0.0:9090->9090/tcp, :::9090->9090/tcp                             order-management2
34fda9942b7c   profile-management     "docker-entrypoint.s…"   6 weeks ago      Up 14 hours               0.0.0.0:3003->3003/tcp, :::3003->3003/tcp                             profile-management2
95ae27901561   mysql:5.7              "docker-entrypoint.s…"   7 months ago     Restarting (1) 21 seconds ago                                                                   mysql
```

4001:4001 order-management

`cat Dockerfile | grep -i "Expose"`

```
mahendrakumar-v@thinkpad-e14-gen-3:~/Study/Microservie_App/Microservice_Project/ecommerce-ui$ cat Dockerfile | grep -i "EXPOSE"
# Expose the port that the server listens on
EXPOSE 4000
mahendrakumar-v@thinkpad-e14-gen-3:~/Study/Microservie_App/Microservice_Project/ecommerce-ui$
```

root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker run -d --name Expose3 --publish-all order-management

```
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker run -d --name Expose3 --publish-all  order-management
2a6e37d1d7a3f3c7c7bf1c33b92821f0da32b5bc8bac68a22c74f994a0953f
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED          STATUS                    PORTS                                                        NAMES
2a6e37d1d7a3   order-management       "/usr/local/bin/mvn-…"   6 seconds ago    Up 5 seconds              0.0.0.0:32768->9090/tcp, [::]:32768->9090/tcp                Expose3
8fdf4a7bf1c3   order-management       "/usr/local/bin/mvn-…"   6 minutes ago    Up 6 minutes              0.0.0.0:4001->4001/tcp, :::4001->4001/tcp, 9090/tcp          Expose2
20195c665396   ecommerce-ui           "docker-entrypoint.s…"   6 weeks ago      Up 14 hours               0.0.0.0:4000->4000/tcp, :::4000->4000/tcp                    ecommerce-ui2
def35cf22b4e   product-inventory      "python3 inventory_a…"   6 weeks ago      Up 14 hours               0.0.0.0:3002->3002/tcp, :::3002->3002/tcp                    product-inventory2
ccc840f8d96d   product-catalog        "docker-entrypoint.s…"   6 weeks ago      Up 14 hours               0.0.0.0:3001->3001/tcp, :::3001->3001/tcp                    product-catalog2
b33cd333983e   contact-support-team   "python3 server.py"      6 weeks ago      Up 14 hours               0.0.0.0:8000->8000/tcp, :::8000->8000/tcp                    contact-support-team2
4a6f60b2d075   order-management       "/usr/local/bin/mvn-…"   6 weeks ago      Up 14 hours               0.0.0.0:9090->9090/tcp, :::9090->9090/tcp                    order-management2
34fda9942b7c   profile-management     "docker-entrypoint.s…"   6 weeks ago      Up 14 hours               0.0.0.0:3003->3003/tcp, :::3003->3003/tcp                    profile-management2
95ae27901561   mysql:5.7              "docker-entrypoint.s…"   7 months ago     Restarting (1) 7 seconds ago                                                               mysql
```

8. How will you check CPU/Mem usage of the container?

We can check docker container CPU and Memory usage by using docke stats command.

`docker stats – Real time Monitoring`

```
CONTAINER ID   NAME                   CPU %    MEM USAGE / LIMIT     MEM %    NET I/O           BLOCK I/O          PIDS
2a6e37d1d7a3   Expose3                0.29%    380.9MiB / 14.46GiB   2.57%    51.6MB / 1.05MB   1.67MB / 63MB      75
8fdf4a7bf1c3   Expose2                0.27%    432.2MiB / 14.46GiB   2.92%    51.6MB / 1.1MB    30MB / 61MB        75
20195c665396   ecommerce-ui2          0.00%    14.76MiB / 14.46GiB  0.10%    584kB / 397kB     14.7MB / 0B        11
def35cf22b4e   product-inventory2     0.02%    32.82MiB / 14.46GiB  0.22%    581kB / 0B        20.5MB / 0B        1
ccc840f8d96d   product-catalog2       0.00%    10.59MiB / 14.46GiB  0.07%    581kB / 0B        7.64MB / 0B        7
b33cd333983e   contact-support-team2  0.01%    32.35MiB / 14.46GiB  0.22%    581kB / 0B        19.5MB / 0B        1
4a6f60b2d075   order-management2      0.26%    392.4MiB / 14.46GiB  2.65%    581kB / 0B        92.1MB / 87.4MB    70
34fda9942b7c   profile-management2    0.00%    12.54MiB / 14.46GiB  0.08%    581kB / 0B        25.5MB / 0B        7
95ae27901561   mysql                  0.00%    0B / 0B              0.00%    0B / 0B           0B / 0B            0
```

`Chekc CPU and Menory for specific container`

```
CONTAINER ID   NAME       CPU %    MEM USAGE / LIMIT     MEM %    NET I/O           BLOCK I/O         PIDS
2a6e37d1d7a3   Expose3    0.30%    381.1MiB / 14.46GiB   2.57%    51.6MB / 1.05MB   1.67MB / 64MB     75
```

CPU% - Cpu Percentage Usage.

MEM USAGE / LIMIT – Memory used and Allocated Memory

MEM % - Percentage of Total Memory Usage

NET IO – Network Traffic

BLOCK IO – Disk Read / Write

PIDS -No of process running.

docker top – Process – level info

9. How will you reduce your size of the docker image?

You can Optimize your docker image size that improve performace, Deployment speed, and Security.

1> Use minimal base image

FROM alpine:latest

2> Use multi stage docker build.

Stage 1: Build

FROM node:18-alpine AS builder

WORKDIR /app

COPY package.json package-lock.json ./

RUN npm install

COPY . .

RUN npm run build

Stage 2: Production Image (only final build artifacts)

FROM node:18-alpine

WORKDIR /app

COPY --from=builder /app/dist ./dist

COPY --from=builder /app/node_modules ./node_modules

CMD ["node", "dist/index.js"]

3> Minimize layer by combining commands.

RUN apt-get update && \

   apt-get install -y curl vim && \

   apt-get clean && rm -rf /var/lib/apt/lists/*

4>Remove unnessary files

.dockerignore file to execute unnessary files.

node_modules

.git

*.log

.env

Dockerfile

5> Use –no-cache and remove unused package.

Docker build –no-cache -t myimage .

RUN apk add --no-cache curl && rm -rf /var/cache/apk/* - After installing Dependancies remove un necessary files.

6> Use minimal Depedancies

FROM scratch

COPY myapp /myapp

CMD ["/myapp"]

7> Use docker image prune – used to clear unused images.containers, Volumes, Networks.

docker system prune -a

10. How will you secure your docker images & containers?

We can secure our docker imags and container is critical to prevent

Vulnerabilities

Un authorized access

Malicious attack

Secure your docker images:

Use official and Trusted images : Pull images from the trusted source (Docker hub, Verified Repo)

docker trust inspect --pretty <image> : Check image signature before using them.

```
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker trust inspect --pretty httpd
time="2025-03-22T01:03:19+05:30" level=warning msg="certificate with CN docker.io/library/httpd is near expiry"
time="2025-03-22T01:03:23+05:30" level=warning msg="certificate with CN docker.io/library/httpd is near expiry"
time="2025-03-22T01:03:23+05:30" level=warning msg="certificate with CN docker.io/library/httpd is near expiry"
time="2025-03-22T01:03:24+05:30" level=warning msg="certificate with CN docker.io/library/httpd is near expiry"
time="2025-03-22T01:03:24+05:30" level=warning msg="certificate with CN docker.io/library/httpd is near expiry"


Signatures for httpd

SIGNED TAG       DIGEST                                                              SIGNERS
2                391a8eb0c1ed464163da46099606a5ec293705118f3054d6c60f5957e2485bd0    (Repo Admin)
2-alpine         4aec2953509e2d3aa5a8d73c580a381be44803fd2481875b15d9ad7d2810d7ca    (Repo Admin)
2-alpine3.13     b0fca692af92787bfd21ccaad1e5b7bea723ca4009dda4eb8018101b9c47cad2    (Repo Admin)
2-alpine3.14     c9bc2ff8e47b7a8cd60c9790eac5e16a847c55c3398862a4f5a1e6116a9748c2    (Repo Admin)
2-alpine3.15     4eb4177b9245c686696dd8120c79cd64b7632b27d890db4cad3b0e844ed737af    (Repo Admin)
2-alpine3.16     4658c554fe215c8a17d57adbd9f8595e4922abda40d1e67a276456d2f142400e    (Repo Admin)
2-alpine3.17     03c154f29d68648c335a19c8bfca1562251bc8af534e10c6f87361551c381d21    (Repo Admin)
2-alpine3.18     b0fc22603ede0b405bbb5ed6d87acae61ebab28d2aca894488b2b0ad6d3edfab    (Repo Admin)
2-alpine3.19     68f6714169b70ca33126c46f60cc802eb602c6435f4ebf08349d871d500d3e70    (Repo Admin)
2-alpine3.20     b64b5734fbc0fbb8fb995d5cc29a2ff2d86ed4c83dfd4f4d82d183f2a66daed4    (Repo Admin)
2-alpine3.21     4aec2953509e2d3aa5a8d73c580a381be44803fd2481875b15d9ad7d2810d7ca    (Repo Admin)
2-bookworm       391a8eb0c1ed464163da46099606a5ec293705118f3054d6c60f5957e2485bd0    (Repo Admin)
2-bullseye       4bdb49b01ab1a7b019410337588f772eebe6ae01d038987aa849fcb798fbf26f    (Repo Admin)
2-buster         f03a63735d3653045a3b1f5490367415e9534d8abfe0b21252454dc85ca09800    (Repo Admin)
2.2              9784d70c8ea466fabd52b0bc8cde84980324f9612380d22fbad2151df9a430eb    (Repo Admin)
2.2-alpine       b2dbea238e355c3a45eb54f2aa4a288b34caa21e0976e6779b6bc85a1fea1097    (Repo Admin)
2.2.29           0a39699d267aaee04382c6b1b4fe2fc30737450fe8d4fabd88eee1a3e0016144    (Repo Admin)
2.2.31           f3f1dfb2b45b6a4fb90f4644a505262f631a18f83c3acf7d426621b592a1411f    (Repo Admin)
2.2.31-alpine    0d0bd21c1ac80e75732ad89a0db4640886431ecd443b3f94e7b441fee40ce927    (Repo Admin)
2.2.32           a28a1a54ee7c4e03249b5eb3fed0c387399ffa5bb422ab50cbb19ffde76e58e7    (Repo Admin)
2.2.32-alpine    274130e43cfb7b54423a4a56af3f6fdec60e2d0dee5cbe1058fb31809d588839    (Repo Admin)
2.2.34           9784d70c8ea466fabd52b0bc8cde84980324f9612380d22fbad2151df9a430eb    (Repo Admin)
2.2.34-alpine    b2dbea238e355c3a45eb54f2aa4a288b34caa21e0976e6779b6bc85a1fea1097    (Repo Admin)
2.4              391a8eb0c1ed464163da46099606a5ec293705118f3054d6c60f5957e2485bd0    (Repo Admin)
2.4-alpine       4aec2953509e2d3aa5a8d73c580a381be44803fd2481875b15d9ad7d2810d7ca    (Repo Admin)
2.4-alpine3.13   b0fca692af92787bfd21ccaad1e5b7bea723ca4009dda4eb8018101b9c47cad2    (Repo Admin)
2.4-alpine3.14   c9bc2ff8e47b7a8cd60c9790eac5e16a847c55c3398862a4f5a1e6116a9748c2    (Repo Admin)
2.4-alpine3.15   4eb4177b9245c686696dd8120c79cd64b7632b27d890db4cad3b0e844ed737af    (Repo Admin)
2.4-alpine3.16   4658c554fe215c8a17d57adbd9f8595e4922abda40d1e67a276456d2f142400e    (Repo Admin)
2.4-alpine3.17   03c154f29d68648c335a19c8bfca1562251bc8af534e10c6f87361551c381d21    (Repo Admin)
2.4-alpine3.18   b0fc22603ede0b405bbb5ed6d87acae61ebab28d2aca894488b2b0ad6d3edfab    (Repo Admin)
2.4-alpine3.19   68f6714169b70ca33126c46f60cc802eb602c6435f4ebf08349d871d500d3e70    (Repo Admin)
2.4-alpine3.20   b64b5734fbc0fbb8fb995d5cc29a2ff2d86ed4c83dfd4f4d82d183f2a66daed4    (Repo Admin)
2.4-alpine3.21   4aec2953509e2d3aa5a8d73c580a381be44803fd2481875b15d9ad7d2810d7ca    (Repo Admin)
2.4-bookworm     391a8eb0c1ed464163da46099606a5ec293705118f3054d6c60f5957e2485bd0    (Repo Admin)
2.4-bullseye     4bdb49b01ab1a7b019410337588f772eebe6ae01d038987aa849fcb798fbf26f    (Repo Admin)
2.4-buster       f03a63735d3653045a3b1f5490367415e9534d8abfe0b21252454dc85ca09800    (Repo Admin)
2.4.10           8efedca9b862c9615c2c411f8e8426e011706233b3c613ba8a357006b835e33a    (Repo Admin)
2.4.12           6d54b079cb30820fb7d185defd5a7cc730385591eb95b1edbced2389f7e73b52    (Repo Admin)
2.4.16           c725ac3661a472b54e10bfb02bb4c490c00c29d0979357fb82fb6f7bb9788999    (Repo Admin)
2.4.17           ff734d5b9bfc1d33936211a9564e1811665ee9487d7ab38732ae38180dd613b9    (Repo Admin)
2.4.18           bcfcee1fe86942a96c0b1ee9c2f1aaaca9c5d0f9d261ed48dff50f41ad31e1ac    (Repo Admin)
2.4.20           ba28a514d23e9e77e2e40a6e0ab0c88d9b43b7fe85388ff3d9a528d7742347eb    (Repo Admin)
```

Avoid using :Latest tag as it may introduce unexpected updates.

You can use insted of latest FROM nagios:1.21-alpine

Scan docker images for vulnerability:

trivy image <Image name>

root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# trivy image order-management

```
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# trivy image order-management
2025-03-22T01:00:48+05:30     INFO    Need to update DB
2025-03-22T01:00:48+05:30     INFO    Downloading DB...      repository="ghcr.io/aquasecurity/trivy-db:2"
61.21 MiB / 61.21 MiB [----------------------------------------------------------------------------------] 100.00% 4.27 MiB p/s 15s
2025-03-22T01:14+05:30        INFO    Vulnerability scanning is enabled
2025-03-22T01:14+05:30        INFO    Secret scanning is enabled
2025-03-22T01:14+05:30        INFO    If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2025-03-22T01:14+05:30        INFO    Please see also https://aquasecurity.github.io/trivy/v0.52/docs/scanner/secret/#recommendation for faster secret detection
2025-03-22T01:17+05:30        FATAL   Fatal error   image scan error: scan error: unable to initialize a scanner: unable to initialize an image scanner: unable to find the specified image "order-management"
in ["docker" "containerd" "podman" "remote"]: 4 errors occurred:
        * docker error: unable to inspect the image (order-management): permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.soc
k/v1.45/images/order-management/json": dial unix /var/run/docker.sock: connect: permission denied
        * containerd error: failed to parse image reference: order-management
        * podman error: unable to initialize Podman client: no podman socket found: stat /run/user/0/snap.trivy/podman/podman.sock: no such file or directory
        * remote error: GET https://index.docker.io/v2/library/order-management/manifests/latest: UNAUTHORIZED: authentication required; [map[Action:pull Class: Name:library/order-management Type:repository]]
```

Remove Sensitive data from the image:

❌ Bad Example: Storing Secrets in Image

RUN echo "DB_PASSWORD=SuperSecret" > /app/config.env

✅ Secure Alternative: Use ENV variables at runtime

ENV DB_PASSWORD=${DB_PASSWORD}

Use least privilege principle (--user flag)

In Docker file

RUN adduser -D myuser

USER myuser

docker run --user 1001 -d order-management

```
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker run --user 1001 -d order-management
71343d0833c33c91173e4724ecfd53ae1ed21b7a9e01dbc93510e1e147b0d58d
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED        STATUS                     PORTS                                               NAMES
2a6e37d1d7a3   order-management      "/usr/local/bin/mvn-…"   45 minutes ago Up 45 minutes              0.0.0.0:32768->9090/tcp, [::]:32768->9090/tcp       Expose3
8fdf4a7bf1c3   order-management      "/usr/local/bin/mvn-…"   51 minutes ago Up 51 minutes              0.0.0.0:4001->4001/tcp, :::4001->4001/tcp, 9090/tcp Expose2
2019Sc66S396   ecommerce-ui          "docker-entrypoint.s…"   6 weeks ago    Up 15 hours                0.0.0.0:4000->4000/tcp, :::4000->4000/tcp           ecommerce-ui2
def35cf22b4e   product-inventory     "python3 inventory_a…"   6 weeks ago    Up 15 hours                0.0.0.0:3002->3002/tcp, :::3002->3002/tcp           product-inventory2
ccc840f8d96d   product-catalog       "docker-entrypoint.s…"   6 weeks ago    Up 15 hours                0.0.0.0:3001->3001/tcp, :::3001->3001/tcp           product-catalog2
b33cd333983e   contact-support-team  "python3 server.py"      6 weeks ago    Up 15 hours                0.0.0.0:8000->8000/tcp, :::8000->8000/tcp           contact-support-team2
4a6f60b2d075   order-management      "/usr/local/bin/mvn-…"   6 weeks ago    Up 15 hours                0.0.0.0:9090->9090/tcp, :::9090->9090/tcp           order-management2
34fda9942b7c   profile-management    "docker-entrypoint.s…"   6 weeks ago    Up 15 hours                0.0.0.0:3003->3003/tcp, :::3003->3003/tcp           profile-management2
95ae27901561   mysql:5.7             "docker-entrypoint.s…"   7 months ago   Restarting (1) 46 seconds ago                                                  mysql
```

root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker inspect 71343d0833c33c91173e4724ecfd53ae1ed21b7a9e01dbc93510e1e147b0d58d | grep -i "user"

```
            "IPPrefixLen": 0,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "DNSNames": null
        }
    }
}
]
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker inspect 71343d0833c33c91173e4724ecfd53ae1ed21b7a9e01dbc93510e1e147b0d58d | grep -i "user"
            "UsersnsMode": "",
            "User": "1001",
```

docker run –privileged=false order-management

```
            ReadOnlyPaths": [
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker inspect 71343d0833c33c91173e4724ecfd53ae1ed21b7a9e01dbc93510e1e147b0d58d | grep -i "privi"
            "Privileged": false,
```

docker run –read-only -d order-management

```
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker inspect 71343d0833c33c91173e4724ecfd53ae1ed21b7a9e01dbc93510e1e147b0d58d | grep -i "read"
            "OomScoreAdj": 0,
            "ReadonlyRootfs": false,
            "BlkioDeviceReadBps": [],
            "BlkioDeviceReadIOps": [],
            "ReadonlyPaths": [
```

11. Explain some security practices for your docker images and containers?

We can secure our docker imags and container is critical to prevent

Vulnerabilities

Un authorized access

Malicious attack

Secure your docker images:

Use official and Trusted images : Pull images from the trusted source (Docker hub, Verified Repo)

docker trust inspect --pretty <image> : Check image signature before using them.

```
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker trust inspect --pretty httpd
time="2025-03-22T01:03:19+05:30" level=warning msg="certificate with CN docker.io/library/httpd is near expiry"
time="2025-03-22T01:03:23+05:30" level=warning msg="certificate with CN docker.io/library/httpd is near expiry"
time="2025-03-22T01:03:23+05:30" level=warning msg="certificate with CN docker.io/library/httpd is near expiry"
time="2025-03-22T01:03:24+05:30" level=warning msg="certificate with CN docker.io/library/httpd is near expiry"
time="2025-03-22T01:03:24+05:30" level=warning msg="certificate with CN docker.io/library/httpd is near expiry"

Signatures for httpd

SIGNED TAG         DIGEST                                                               SIGNERS
2                  391a8eb0c1ed464163da46099606a5ec293705118f3054d6c60f5957e2485bd0     (Repo Admin)
2-alpine           4aec2953509e2d3aa5a8d73c580a381be44803fd2481875b15d9ad7d2810d7ca     (Repo Admin)
2-alpine3.13       b0fca692af92787bfd21ccaad1e5b7bea723ca4009dda4eb8018101b9c47cad2     (Repo Admin)
2-alpine3.14       c9bc2ff8e47b7a8cd60c9790eac5e16a847c55c3398862a4f5a1e6116a9748c2     (Repo Admin)
2-alpine3.15       4eb4177b9245c686696dd8120c79cd64b7632b27d890db4cad3b0e844ed737af     (Repo Admin)
2-alpine3.16       4658c554fe215c8a17d57adbd9f8595e4922abda40d1e67a276456d2f142400e     (Repo Admin)
2-alpine3.17       03c154f29d68648c335a19c8bfca1562251bc8af534e10c6f87361551c381d21     (Repo Admin)
2-alpine3.18       b0fc22603ede0b405bbb5ed6d87acae61ebab28d2aca894488b2b0ad6d3edfab     (Repo Admin)
2-alpine3.19       68f6714169b70ca33126c46f60cc802eb602c6435f4ebf08349d871d500d3e70     (Repo Admin)
2-alpine3.20       b64b5734fbc0fbb8fb995d5cc29a2ff2d86ed4c83dfd4f4d82d183f2a66daed4     (Repo Admin)
2-alpine3.21       4aec2953509e2d3aa5a8d73c580a381be44803fd2481875b15d9ad7d2810d7ca     (Repo Admin)
2-bookworm         391a8eb0c1ed464163da46099606a5ec293705118f3054d6c60f5957e2485bd0     (Repo Admin)
2-bullseye         4bdb49b01ab1a7b019410337588f772eebe6ae01d038987aa849fcb798fbf26f     (Repo Admin)
2-buster           f03a63735d3653045a3b1f5490367415e9534d8abfe0b21252454dc85ca09800     (Repo Admin)
2.2                9784d70c8ea466fabd52b0bc8cde84980324f9612380d22fbad2151df9a430eb     (Repo Admin)
2.2-alpine         b2dbea238e355c3a45eb54f2aa4a288b34caa21e0976e6779b6bc85a1fea1097     (Repo Admin)
2.2.29             0a39699d267aaee04382c6b1b4fe2fc30737450fe8d4fabd88eee1a3e0016144     (Repo Admin)
2.2.31             f3f1dfb2b45b6a4fb90f4644a505262f631a18f83c3acf7d426621b592a1411f     (Repo Admin)
2.2.31-alpine      0d0bd21c1ac80e75732ad89a0db4640886431ecd443b3f94e7b441fee40ce927     (Repo Admin)
2.2.32             a28a1a54ee7c4e03249b5eb3fed0c387399ffa5bb422ab50cbb19ffde76e58e7     (Repo Admin)
2.2.32-alpine      274130e43cfb7b54423a4a56af3f6fdec60e2d0dee5cbe1058fb31809d588839     (Repo Admin)
2.2.34             9784d70c8ea466fabd52b0bc8cde84980324f9612380d22fbad2151df9a430eb     (Repo Admin)
2.2.34-alpine      b2dbea238e355c3a45eb54f2aa4a288b34caa21e0976e6779b6bc85a1fea1097     (Repo Admin)
2.4                391a8eb0c1ed464163da46099606a5ec293705118f3054d6c60f5957e2485bd0     (Repo Admin)
2.4-alpine         4aec2953509e2d3aa5a8d73c580a381be44803fd2481875b15d9ad7d2810d7ca     (Repo Admin)
2.4-alpine3.13     b0fca692af92787bfd21ccaad1e5b7bea723ca4009dda4eb8018101b9c47cad2     (Repo Admin)
2.4-alpine3.14     c9bc2ff8e47b7a8cd60c9790eac5e16a847c55c3398862a4f5a1e6116a9748c2     (Repo Admin)
2.4-alpine3.15     4eb4177b9245c686696dd8120c79cd64b7632b27d890db4cad3b0e844ed737af     (Repo Admin)
2.4-alpine3.16     4658c554fe215c8a17d57adbd9f8595e4922abda40d1e67a276456d2f142400e     (Repo Admin)
2.4-alpine3.17     03c154f29d68648c335a19c8bfca1562251bc8af534e10c6f87361551c381d21     (Repo Admin)
2.4-alpine3.18     b0fc22603ede0b405bbb5ed6d87acae61ebab28d2aca894488b2b0ad6d3edfab     (Repo Admin)
2.4-alpine3.19     68f6714169b70ca33126c46f60cc802eb602c6435f4ebf08349d871d500d3e70     (Repo Admin)
2.4-alpine3.20     b64b5734fbc0fbb8fb995d5cc29a2ff2d86ed4c83dfd4f4d82d183f2a66daed4     (Repo Admin)
2.4-alpine3.21     4aec2953509e2d3aa5a8d73c580a381be44803fd2481875b15d9ad7d2810d7ca     (Repo Admin)
2.4-bookworm       391a8eb0c1ed464163da46099606a5ec293705118f3054d6c60f5957e2485bd0     (Repo Admin)
2.4-bullseye       4bdb49b01ab1a7b019410337588f772eebe6ae01d038987aa849fcb798fbf26f     (Repo Admin)
2.4-buster         f03a63735d3653045a3b1f5490367415e9534d8abfe0b21252454dc85ca09800     (Repo Admin)
2.4.10             8efedca9b862c9615c2c411f8e8426e011706233b3c613ba8a357006b835e33a     (Repo Admin)
2.4.12             6d54b079cb30820fb7d185defd5a7cc730385591eb95b1edbced2389f7e73b52     (Repo Admin)
2.4.16             c725ac3661a472b54e10bfb02bb4c490c00c29d0979357fb82fb6f7bb9788999     (Repo Admin)
2.4.17             ff734d5b9bfc1d33936211a9564e1811665ee9487d7ab38732ae38180dd613b9     (Repo Admin)
2.4.18             bcfcee1fe86942a96c0b1ee9c2f1aaaca9c5d0f9d261ed48dff50f41ad31e1ac     (Repo Admin)
2.4.20             ba28a514d23e9e77e2e40a6e0ab0c88d9b43b7fe85388ff3d9a528d7742347eb     (Repo Admin)
```

Avoid using :Latest tag as it may introduce unexpected updates.

You can use insted of latest FROM nagios:1.21-alpine

==Scan docker images for vulnerability:==

trivy image <Image name>

root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# trivy image order-management

```
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# trivy image order-management
2025-03-22T01:00:48+05:30    INFO    Need to update DB
2025-03-22T01:00:48+05:30    INFO    Downloading DB...    repository="ghcr.io/aquasecurity/trivy-db:2"
61.21 MiB / 61.21 MiB [------------------------------------------------------------------------] 100.00% 4.27 MiB p/s 15s
2025-03-22T01:01:14+05:30    INFO    Vulnerability scanning is enabled
2025-03-22T01:01:14+05:30    INFO    Secret scanning is enabled
2025-03-22T01:01:14+05:30    INFO    If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2025-03-22T01:01:14+05:30    INFO    Please see also https://aquasecurity.github.io/trivy/v0.52/docs/scanner/secret#recommendation for faster secret detection
2025-03-22T01:01:17+05:30    FATAL   Fatal error    image scan error: scan error: unable to initialize a scanner: unable to initialize an image scanner: unable to find the specified image "order-management"
in ["docker" "containerd" "podman" "remote"]: 4 errors occurred:
    * docker error: unable to inspect the image (order-management): permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.soc
k/v1.45/images/order-management/json": dial unix /var/run/docker.sock: connect: permission denied
    * containerd error: failed to parse image reference: order-management
    * podman error: unable to initialize Podman client: no podman socket found: stat /run/user/0/snap.trivy/podman/podman.sock: no such file or directory
    * remote error: GET https://index.docker.io/v2/library/order-management/manifests/latest: UNAUTHORIZED: authentication required; [map[Action:pull Class: Name:library/order-management Type:repository]]
```

==Remove Sensitive data from the image:==

❌ Bad Example: Storing Secrets in Image

RUN echo "DB_PASSWORD=SuperSecret" > /app/config.env

✅ Secure Alternative: Use ENV variables at runtime

ENV DB_PASSWORD=${DB_PASSWORD}

Use least privilege principle (--user flag)

In Docker file

RUN adduser -D myuser

USER myuser

docker run --user 1001 -d order-management

```
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker run --user 1001 -d order-management
71343d0833c33c91173e4724ecfd53ae1ed21b7a9e01dbc93510e1e147b0d58d
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker ps
CONTAINER ID   IMAGE                 COMMAND                CREATED          STATUS                    PORTS                                              NAMES
2a6e37d1d7a3   order-management      "/usr/local/bin/mvn-…" 45 minutes ago   Up 45 minutes             0.0.0.0:32768->9090/tcp, [::]:32768->9090/tcp      Expose3
8fdf4a7bf1c3   order-management      "/usr/local/bin/mvn-…" 51 minutes ago   Up 51 minutes             0.0.0.0:4001->4001/tcp, :::4001->4001/tcp, 9090/tcp Expose2
20195c665396   ecommerce-ui          "docker-entrypoint.s…" 6 weeks ago      Up 15 hours               0.0.0.0:4000->4000/tcp, :::4000->4000/tcp          ecommerce-ui2
def35cf22b4e   product-inventory     "python3 inventory_a…" 6 weeks ago      Up 15 hours               0.0.0.0:3002->3002/tcp, :::3002->3002/tcp          product-inventory2
ccc840f8d96d   product-catalog       "docker-entrypoint.s…" 6 weeks ago      Up 15 hours               0.0.0.0:3001->3001/tcp, :::3001->3001/tcp          product-catalog2
b33cd333983e   contact-support-team  "python3 server.py"    6 weeks ago      Up 15 hours               0.0.0.0:8000->8000/tcp, :::8000->8000/tcp          contact-support-team2
4a6f60b2d075   order-management      "/usr/local/bin/mvn-…" 6 weeks ago      Up 15 hours               0.0.0.0:9090->9090/tcp, :::9090->9090/tcp          order-management2
34fda9942b7c   profile-management    "docker-entrypoint.s…" 6 weeks ago      Up 15 hours               0.0.0.0:3003->3003/tcp, :::3003->3003/tcp          profile-management2
95ae27901561   mysql:5.7             "docker-entrypoint.s…" 7 months ago     Restarting (1) 46 seconds ago                                                mysql
```

root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker inspect 71343d0833c33c91173e4724ecfd53ae1ed21b7a9e01dbc93510e1e147b0d58d | grep -i "user"

```
            "IPPrefixLen": 0,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "DNSNames": null
        }
    }
  }
}
]
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker inspect 71343d0833c33c91173e4724ecfd53ae1ed21b7a9e01dbc93510e1e147b0d58d | grep -i "user"
            "UsernsMode": "",
            "User": "1001",
```

Disable previleged mode:

docker run –privileged=false order-management

```
            ReadOnlyPaths": [
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker inspect 71343d0833c33c91173e4724ecfd53ae1ed21b7a9e01dbc93510e1e147b0d58d | grep -i "privi"
            "Privileged": false,
```

Use read only file system:

docker run –read-only -d order-management

```
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker inspect 71343d0833c33c91173e4724ecfd53ae1ed21b7a9e01dbc93510e1e147b0d58d | grep -i "read"
            "OomScoreAdj": 0,
            "ReadonlyRootfs": false,
            "BlkioDeviceReadBps": [],
            "BlkioDeviceReadIOps": [],
            "ReadonlyPaths": [
```

12. How will you Monitor your docker container in Production?

Monitoring docker container in production is critical for ensuring performance, Stablity and security.

You need to track CPU Usage, Memory, Disk io, Network traffic, logs.

Built in Command:

docker ps

```
            "Privileged": false,
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# docker ps
CONTAINER ID   IMAGE                 COMMAND                CREATED          STATUS                    PORTS                                              NAMES
2a6e37d1d7a3   order-management      "/usr/local/bin/mvn-…" 2 hours ago      Up 2 hours                0.0.0.0:32768->9090/tcp, [::]:32768->9090/tcp      Expose3
8fdf4a7bf1c3   order-management      "/usr/local/bin/mvn-…" 2 hours ago      Up 2 hours                0.0.0.0:4001->4001/tcp, :::4001->4001/tcp, 9090/tcp Expose2
20195c665396   ecommerce-ui          "docker-entrypoint.s…" 6 weeks ago      Up 16 hours               0.0.0.0:4000->4000/tcp, :::4000->4000/tcp          ecommerce-ui2
def35cf22b4e   product-inventory     "python3 inventory_a…" 6 weeks ago      Up 16 hours               0.0.0.0:3002->3002/tcp, :::3002->3002/tcp          product-inventory2
ccc840f8d96d   product-catalog       "docker-entrypoint.s…" 6 weeks ago      Up 16 hours               0.0.0.0:3001->3001/tcp, :::3001->3001/tcp          product-catalog2
b33cd333983e   contact-support-team  "python3 server.py"    6 weeks ago      Up 16 hours               0.0.0.0:8000->8000/tcp, :::8000->8000/tcp          contact-support-team2
4a6f60b2d075   order-management      "/usr/local/bin/mvn-…" 6 weeks ago      Up 16 hours               0.0.0.0:9090->9090/tcp, :::9090->9090/tcp          order-management2
34fda9942b7c   profile-management    "docker-entrypoint.s…" 6 weeks ago      Up 16 hours               0.0.0.0:3003->3003/tcp, :::3003->3003/tcp          profile-management2
95ae27901561   mysql:5.7             "docker-entrypoint.s…" 7 months ago     Restarting (1) 54 seconds ago                                                mysql
root@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui#
```

Monitor Real time CPU and Memory:

docker stats

```
CONTAINER ID   NAME                   CPU %    MEM USAGE / LIMIT     MEM %    NET I/O          BLOCK I/O         PIDS
2a6e37d1d7a3   Expose3                0.18%    391.4MiB / 14.46GiB   2.64%    51.7MB / 1.05MB  1.67MB / 75.2MB   75
8fdf4a7bf1c3   Expose2                0.19%    442MiB / 14.46GiB     2.99%    51.7MB / 1.1MB   30MB / 72.7MB     75
20195c665396   ecommerce-ui2          0.00%    14.76MiB / 14.46GiB   0.10%    659kB / 397kB    14.7MB / 0B       11
def35cf22b4e   product-inventory2     0.01%    19.61MiB / 14.46GiB   0.13%    656kB / 0B       20.5MB / 0B       1
ccc840f8d96d   product-catalog2       0.00%    10.59MiB / 14.46GiB   0.07%    656kB / 0B       7.64MB / 0B       7
b33cd333983e   contact-support-team2  0.01%    19.56MiB / 14.46GiB   0.13%    656kB / 0B       19.5MB / 0B       1
4a6f60b2d075   order-management2      0.20%    349.8MiB / 14.46GiB   2.36%    656kB / 0B       92.1MB / 98.9MB   70
34fda9942b7c   profile-management2    0.00%    12.54MiB / 14.46GiB   0.08%    656kB / 0B       25.5MB / 0B       7
95ae27901561   mysql                  0.00%    0B / 0B               0.00%    0B / 0B          0B / 0B           0
```

docker logs -f <Container name>

docker inspect <Container name>

13. What is docker swarm? Diff btw Docker & Docker Swarm?

Docker swarm is a native cluster tool – Its used to manage group of containers.

Docker swarm multi node architecture

Docker swarm provide the following:

Docker swarm support load balancing

Docker swarm, Overlay network support

Failover automation

Docker swarm doing scaling automatically.

Docker swarm is a multi host app.

scalablity

High availablity

Load Balancing

Support service discovery and rolling Updates

Docker – Is a core technology to enble the creation of container.

Docker is a containerize platform.

Docker is a single node archicture.

Docker not supported load balancing

Docker support bidge, None, host, Overlay networks.

Docker no failover automate

Docker need to do manually scaling.

Docker is single host apps.

14. Diff Btw Docker swarm & Kubernetes?

Docker swarm is a native cluster tool – Its used to manage group of containers.

Docker swarm is a simpler, Light weight, Container orchestration.

Managing docker container accross a cluster.

Use manager and worker node, Tasks are distributed across workers.

Built in load balancing.

Manager node replicating high availablity.

Uses a Overlay for Multi container communiction.

Manual Scaling using docker service scale.

Supports local and shared storage. Limited option.

Docker Swarm limited logging with docker logs.

Kubernetes is a more complex, powerful and versatile platform for orchestration containers.

Especially for large scale cloud native application

Kubernets uses Master and worker nodes.

Ingress and extranel load balancing.

Built in self healing with pod reschduling if node is failed.

Using Docker container network interface with better customization.

Kubernetes auto scaling with horizontal pod auto scaler bashed on cpu and memory.

Multi storage backend (AWS EBS, NSF, Azure Disk)

Kubernets integrate with prometheus, grafana, ELK Stack for monitoring and logging.

Kubernetes	Docker
Kubernetes is an ecosystem for managing a cluster of Docker containers known as Pods.	Docker is a container platform for building, configuring and distributing Docker containers.
Kubernetes is not a complete solution and uses custom plugins to extend its functionality.	Docker uses its very own native clustering solution for Docker containers called Docker Swarm.
Load balancing comes out of the box in Kubernetes because of its architecture and it's very convenient	The load balancer is deployed on its own single node swarm when pods in the container are defined as service.
Takes relatively more time for installation.	Setup is quick and easy compared to Kubernetes.

15. When .dockerignore file is used?

A .dockerignore file is more over similier to .gitignore file.

When creating docker image .dockerignore file to exclude files and directories from docker build.

Reduce image size: Prevent unneccesary file from being added to the image.

Speed up Builds: Avoid copying large or irrelevant file to docker build.

Improve Security: Keep sensitive files out of the image.

Prevent Unwanted Changes: Avoid Unnessary rebuilds caused by Unchanged files.

Create a .dockerignore file in the same directory where the docker file available.

```
oot@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# ls -ltrca
otal 28
rw-r--r--   1 mahendrakumar-v mahendrakumar-v 2768 Jan 10 01:42 README.md
rwxrwxrwx   1 mahendrakumar-v mahendrakumar-v  772 Jan 10 01:42 Dockerfile
rwxr-xr-x   3 mahendrakumar-v mahendrakumar-v 4096 Jan 10 01:42 server
rw-r--r--   1 mahendrakumar-v mahendrakumar-v   20 Jan 10 01:42 .dockerignore
rwxr-xr-x   5 mahendrakumar-v mahendrakumar-v 4096 Jan 10 01:42 client
rwxr-xr-x   4 mahendrakumar-v mahendrakumar-v 4096 Feb 12 11:54 .
rwxr-xr-x  10 mahendrakumar-v mahendrakumar-v 4096 Mar  5 10:47 ..
oot@thinkpad-e14-gen-3:/home/mahendrakumar-v/Study/Microservie_App/Microservice_Project/ecommerce-ui# cat .dockerignore
ockerfile
```

All The Best