```
#Multi Stage Docker file
------------------------
FROM maven:3.8.6-openjdk-11 as build
RUN git clone https://github.com/spring-projects/spring-petclinic.git && \
    cd spring-petclinic && \
    mvn package
# jar location /spring-petclinic/target/spring-petclinic-2.7.3.jar

FROM openjdk:11
LABEL project="petclinic"
LABEL author="devops team"
EXPOSE 8080
COPY --from=build /spring-petclinic/target/spring-petclinic-2.7.3.jar /spring-petclinic-2.7.3.jar
CMD ["java", "-jar", "/spring-petclinic-2.7.3.jar"]
```

Docker Instructions in details:
https://directdevops.blog/2019/09/26/docker-image-creation-and-docker-image-layers/

ADD: copy from local as well as remote (web or http urls or git)
COPY: copy only from local machine
WORKDIR: from which path or from where we can execute

ARG: is used to pass some value for the purpose of image building
ENV: sets the environmental variable with value in the container.

USER: is used to set the default user in the container when the container is started/created.

Storage Drivers allow you to create the data in the writable layer of the container. Docker supports the following storage drivers: overlay2, aufs, Btrfs, Device Mapper, Zfs

What is docker volume?
Docker network is a virtual network infrastructure that allows multiple containers to communicate with each other as if they were on the same physical network. You can also expose specific ports on a container to the outside world, allowing other systems to communicate with that container.
It provides several built-in network drivers that allow you to create different types of networks, such as bridge networks, overlay networks, and macvlan networks.

Replicated Services model: In this case swarm manager distributes a specific number of replica task among the nodes based upon the scale you set in the desired state

Global Services Model: In this case swarm runs one task for the service on every available node in the cluster.

What is service in docker swarm?
A service defines the desired state of a containerized application, including the number of replicas, network settings, storage volumes, and other parameters.

What is docker image?
The Docker image help to create Docker containers. You can create the Docker image with the build command. Docker images are stored in the Docker registry.

What is docker file?
Docker file approach is all about using instructions to inform docker what needs to be done when creating the image, starting the container. What are the metadata about your image.

What is Docker Swarm?
Docker Swarm is a container orchestration tool that allows you to manage a cluster of Docker hosts and deploy applications as a swarm of Docker containers. It provides a native clustering and scheduling functionality for Docker containers, making it easier to manage and scale containerized applications across a cluster of hosts.

Docker 🔗

What is docker?
Docker is an open-source containerization platform that automates the deployment, scaling, and management of applications in containers. Docker provides a consistent, reproducible runtime environment that makes it easier to build, ship, and run applications in any environment.

What is Docker Engine?
Docker daemon or Docker engine represents the server. The docker daemon and the clients should be run on the same or remote host, which can communicate through command-line client binary and full RESTful API.

What is docker container?
A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Docker Volume: are preferred mechanisms for persisting data generated by and used by Docker Containers
Advantages:
1. Easier to backup or migrate that bind mounts
2. Can be managed by Docker CLI or the API
3. Work in both Linux & Windows Containers
4. Volume Drivers lets to store volumes on remote hosts or cloud providers, encryption can be added.
5. Volumes do not increase the size of the docker containers writable layer as the volume's contents exist outside of container lifecycle

What is Docker Compose ?
Docker Compose is a tool that allows you to define and run multi-container Docker applications with all its dependencies, such as databases, web servers, and message brokers, and configure their networking and storage settings. Docker Compose then creates and starts the containers, mounts the volumes, and connects the networks, all with a single command.

What is Docker Networking ?
Docker networking refers to the way that Docker containers can communicate with each other and with other networked resources outside of the container environment.

What are Docker's most notable features?
Application agility, Easy modeling, Operational efficiencies, Placement and affinity, Version control

What are Docker's most notable features?
Doesn't provide a storage option, poor monitoring option, No automatic rescheduling of inactive Nodes, Automatic horizontal scaling set up is complicated

To create container :
docker container run <image-name>

To create container with name and tag :
docker container run --name <cont-name> <image>:<tag>

Bridge networking is the default networking mode in Docker. Containers on the same bridge network can communicate with each other using IP addresses assigned by the Docker daemon. Containers can also communicate with the host machine and with other devices on the same physical network.

Host networking: In this mode, the container shares the host's networking stack and is directly connected to the host network interface. This allows the container to use the host's IP address and network configuration, but it can also introduce security risks.

Overlay networking: This mode enables communication between containers running on different Docker hosts. It uses an overlay network that spans multiple Docker hosts and allows containers to communicate with each other using their container names or service names, regardless of their physical location.

Macvlan networking: This mode creates a virtual interface on the host that is connected to a physical network interface, allowing containers to be directly connected to the physical network with their own MAC addresses.

None networking: In this mode, the container has no access to the network, which can be useful for running containers that do not require network access.

Bridge is the default network:
docker network create -d bridge --subnet "10.10..10.0/24" my_br