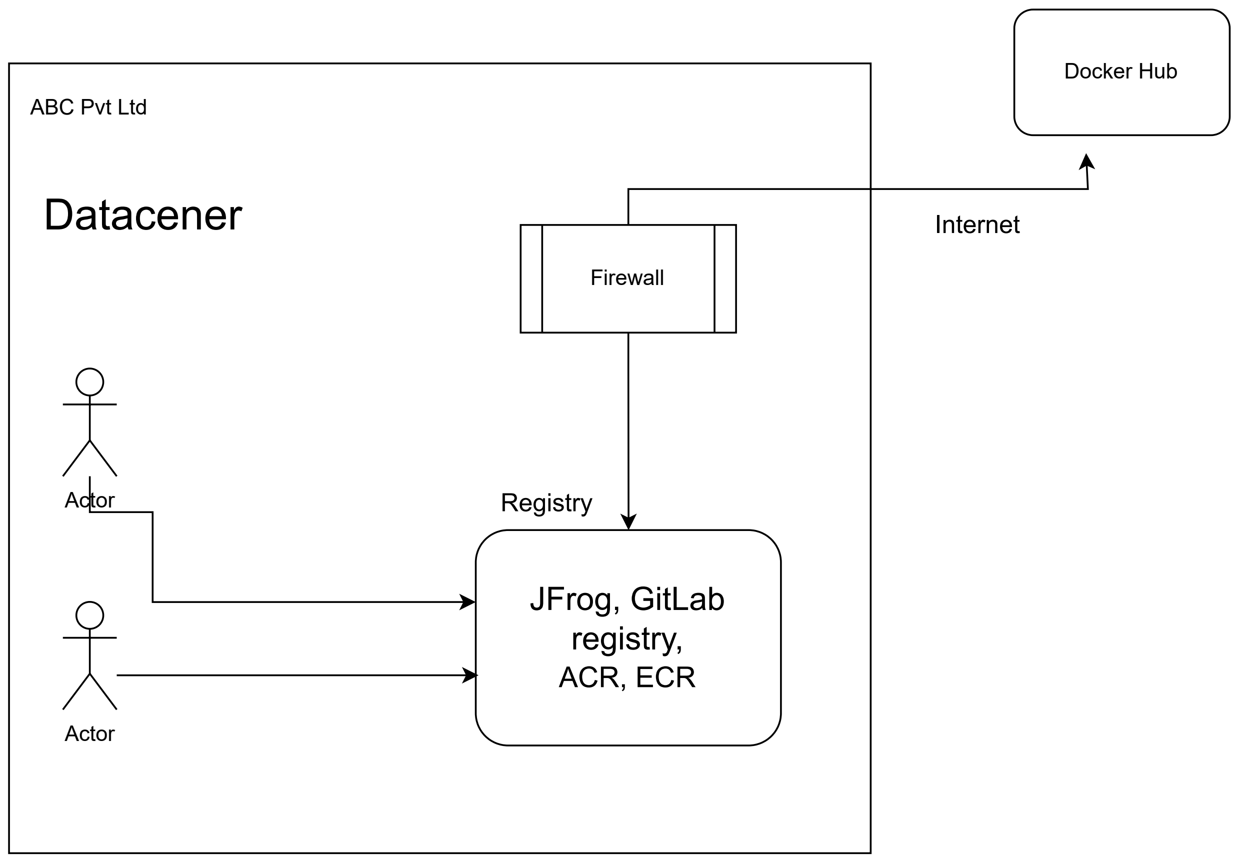
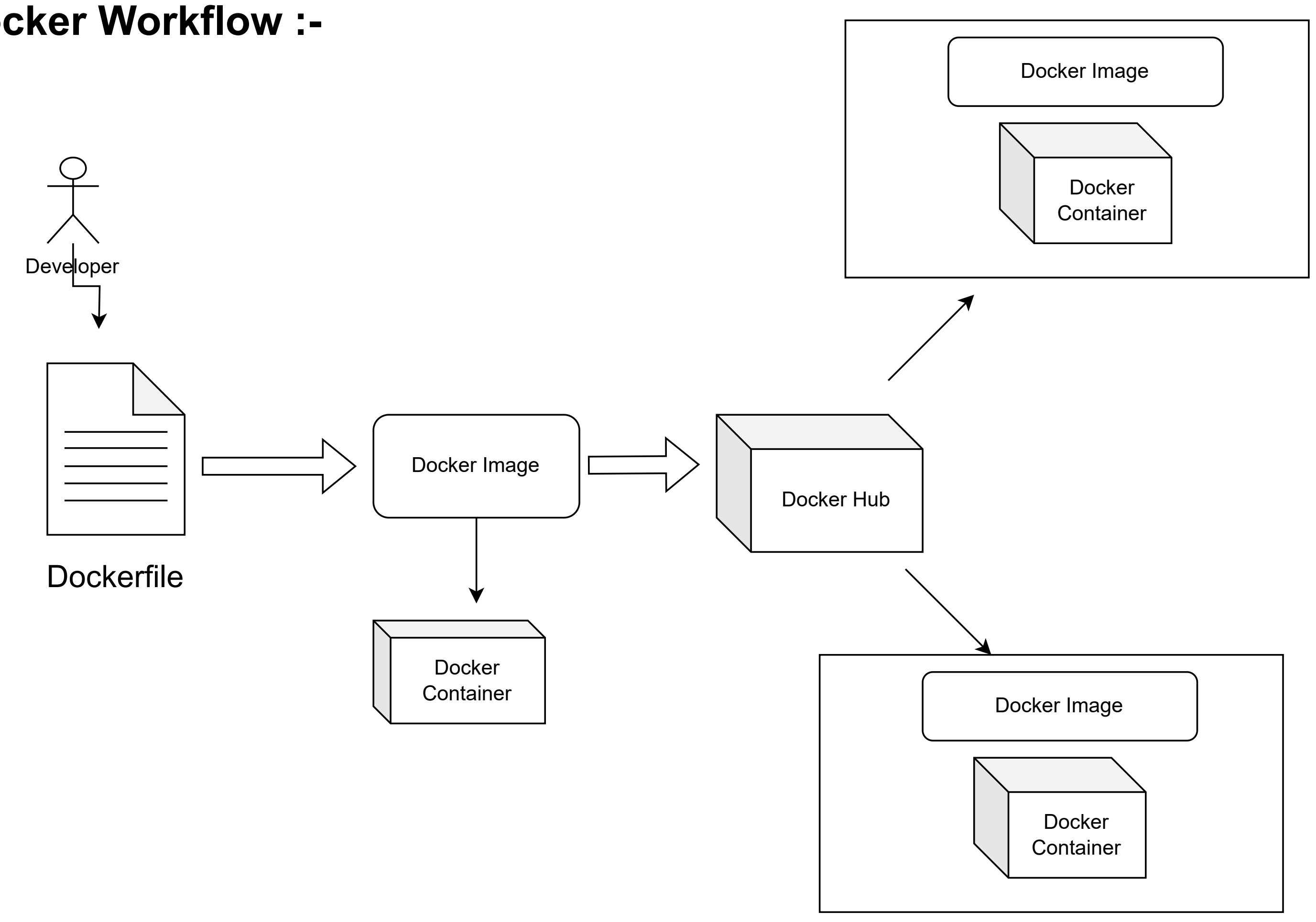


Docker Workflow :-



Docker Volume :-

Mechanism for persistent data storage used by Docker containers

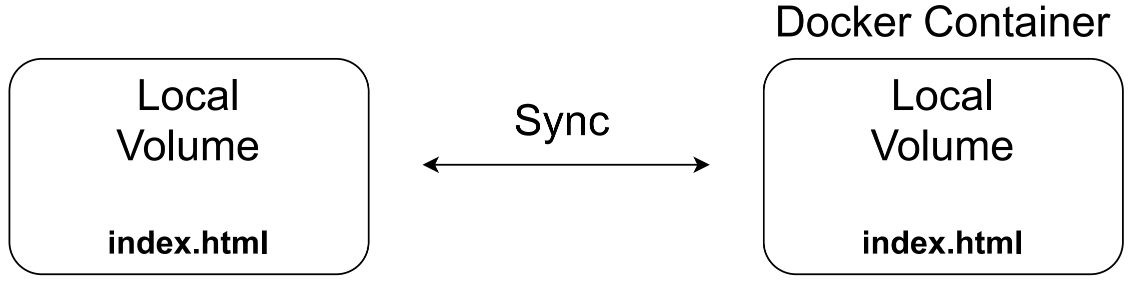
Types of Volumes :-

1. **Volumes:-** Managed by Docker
2. **Bind mounts:-** Depend on the directory structure of the host
3. **Tmpfs mounts:-** Store data in the host system's memory

Docker Volume Commands :-

1. **docker volume ls:-** List all volumes
2. **docker volume create [OPTIONS] VOLUME:** Creates a new volume
3. **docker volume inspect VOLUME:-** Display detailed information
4. **docker volume rm VOLUME:-** Remove a volume
5. **docker run -v VOLUME:/path/in/container:** Mount a volume to a container

Note- If we make any chnages locally, the container will reflect the same changes.
All the local volumes will be persistent throughout but the container will have the same data untill and unless it is killed, the moment it is killed all the data wil be erased.



DockerFile Comman Commands :-

1. **FROM**: Specifies the base image

1. **RUN**: Executes commands in a new layer

1. **CMD**: Provides a default command to run the container

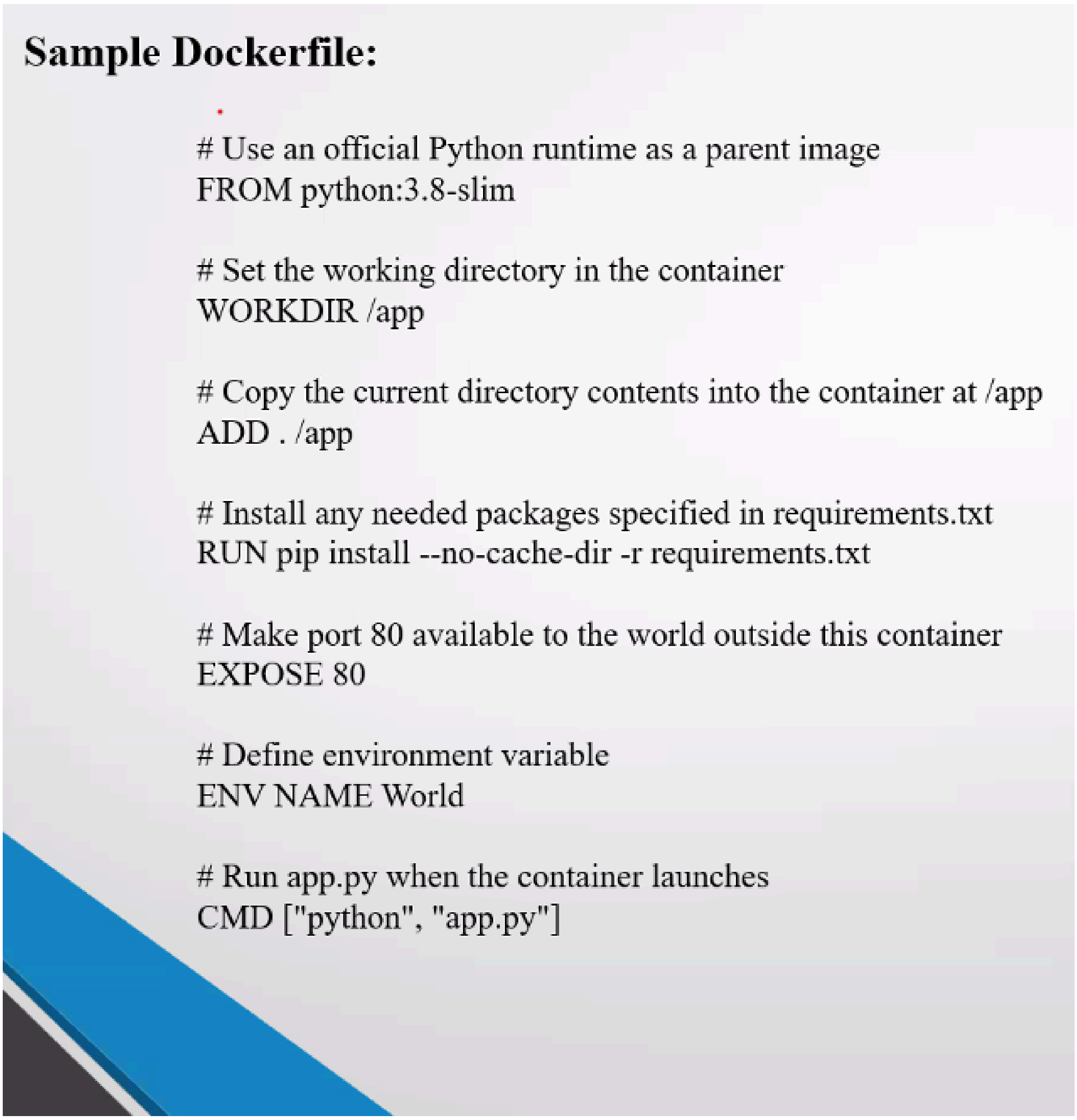
1. **COPY/ADD**: Copies files/directories into the image

1. **EXPOSE**: Expose ports. (e.g.- 80)

1. **ENV**: Sets environment variables

1. **WORKDIR**: Sets the working directory

The above commands are mentioned in a DockerFIle, now we'll see a sample docker file.



Sample Dockerfile:

```
# Use an official Python runtime as a parent image
FROM python:3.8-slim

# Set the working directory in the container
WORKDIR /app

# Copy the current directory contents into the container at /app
ADD . /app

# Install any needed packages specified in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt

# Make port 80 available to the world outside this container
EXPOSE 80

# Define environment variable
ENV NAME World

# Run app.py when the container launches
CMD ["python", "app.py"]
```

Docker Compose :-

A tool for defining and running multi-container Docker applications

Uses a docker-compose.yml file to configure applications services.

Benefits: Simplifies multi-container deployments, manages networks, and volumes

Docker Compose Commands :-

docker-cmpose up: create and start containers

docker-compose down: Stop and remove conatiners, networks, volumes, and images

docker-compose build: Build or rebuild services

docker-compose logs: View output from containers

docker-compose ps: List containers

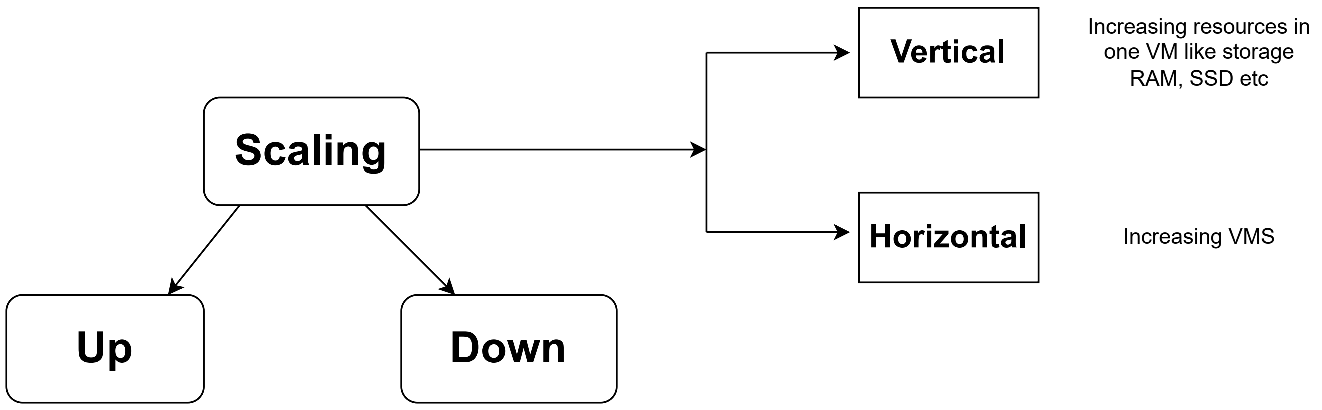
docker-compose stop: stop services.

Introduction to Kubernetes :-

Kubernetes, often referred as **"K8s"**, is an open-source container orchestration platform designed to automate the **deployment, scaling,** and **management** of containerized application.

It was originally developed by **Google** and is now maintained by **Cloud Native computing Foundation(CNCF).**

Kubernetes provides a powerful toolset for managing the complexities of deploying and maintaining containerized application at scale.



Note- Self-healing in Kubernetes is a feature that automatically restores the state of a cluster when it deviates from the desired state. This is done by continuously monitoring the health of the cluster and taking action to restore it if needed.