**Week 10:Docker Compose and installation of Minikube and working with it**

1. Evaluation of previous experiments
2. Hands-on practice of pulling multiple images from DockerHub, running them as single container using DockerCompose.
3. Installation of Minikube, Pulling Nginx image from DockerHUb using minikube.
4. Upload the screenshots for above task

Procedure

**What is Docker Compose?**

Docker Compose is a tool used to define and run multi-container Docker applications. It allows you to define services, networks, and volumes that your application needs, all in a single file. This makes it easier to manage complex applications that require multiple containers (e.g., a web server and a database).

**Structure of a Docker Compose File**

A Docker Compose file is written in YAML (YAML Ain't Markup Language), a human-readable data format used for configuration. In the Docker context, it helps define the services, networks, and volumes needed for an application.

#### ****Basic Components of a Docker Compose YAML File****

1. **Version**: Specifies which version of the Docker Compose file format you're using. Each version comes with its own set of features.

version: '3.8'

1. **Services**: This section defines all the containers you want to run for your application. Each service represents a container (like a web server or database).

Each service includes:

* + **Image**: The Docker image to use.
  + **Ports**: Ports to map between the container and your host machine.
  + **Environment**: Environment variables required by the service.
  + **Dependencies**: Which other services a container depends on.

Example:

services:

web:

image: nginx

ports:

- "8060:80"

db:

image: tomee

ports:

- "8050:8080"

1. **Networks**: Define networks to allow different services to communicate with each other. Docker Compose automatically creates a default network if not specified.
2. **Volumes**: Used for persistent storage, allowing data to persist even after the container stops or is removed.

#### ****Example Docker Compose File (Simple)****

Here’s a basic example of a Docker Compose file that runs WordPress and MySQL together:

version: '3.8' # Docker Compose file format version

services:

wordpress: # WordPress service

image: wordpress:latest

ports:

- "8080:80" # Map port 80 of the container to port 8080 of the host

environment:

WORDPRESS\_DB\_HOST: db:3306 # Database host

WORDPRESS\_DB\_USER: wordpress

WORDPRESS\_DB\_PASSWORD: wordpress

WORDPRESS\_DB\_NAME: wordpress

depends\_on:

- db # Ensures the db service starts first

db: # MySQL service

image: mysql:5.7

environment:

MYSQL\_ROOT\_PASSWORD: rootpassword

MYSQL\_DATABASE: wordpress

MYSQL\_USER: wordpress

MYSQL\_PASSWORD: wordpress

**How to Run It**

To run a multi-container setup like the one above:

1. Save the file as docker-compose.yml.

docker-compose up –d

1. To stop the containers

docker-compose down

1. To scale the container

docker-compose up --scale <service name>=2 -d

### Summary:

* **Docker Compose**: Simplifies managing multi-container applications.
* **YAML**: A readable format used for configuration files, where data is structured in key-value pairs and lists.
* A **docker-compose.yml** file defines services, images, ports, and other configurations to make running multiple containers easier.

Here’s a step-by-step guide to install **Minikube** on Windows:

### ****Step 1: Install Prerequisites****

Before installing Minikube, ensure the following are installed:

1. **Virtualization Support**:
   * Verify virtualization is enabled:
2. **Hypervisor**:
   * Minikube supports multiple hypervisors (e.g., **Hyper-V**, **VirtualBox**, or **Docker** as a driver).
   * Install one of the following:
     + **Hyper-V** (pre-installed on Windows 10/11 Pro or Enterprise).
     + **Docker Desktop** (if you want to use Docker as the driver).

### ****Step 2: Download Minikube****

1. Open a PowerShell or Command Prompt with administrator privileges.
2. Download the latest Minikube executable using this command:
3. curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-installer.exe
4. Install Minikube by running the installer:
5. .\minikube-installer.exe

### ****Step 3: Add Minikube to PATH****

If Minikube is not automatically added to your PATH during installation:

1. Open **System Properties** → **Environment Variables**.
2. Add the directory where Minikube is installed (e.g., C:\Program Files\Minikube) to your PATH variable.

### ****Step 4: Start Minikube****

1. Open a terminal (PowerShell or CMD).
2. Start Minikube with a specified driver (e.g., Hyper-V, Docker, or VirtualBox). For example:
   * **Hyper-V**:
   * minikube start --driver=hyperv
   * **Docker**:
   * minikube start --driver=docker
3. Verify Minikube is running:
4. minikube status

### ****Step 5: Interact with Minikube****

Once Minikube is running:

1. Use kubectl to interact with the cluster.
   * Install kubectl if not already installed:
   * minikube kubectl -- get pods -A
   * Or download it separately from the [official Kubernetes site](https://kubernetes.io/docs/tasks/tools/install-kubectl/).
2. Open the Minikube dashboard (optional):
3. minikube dashboard

### ****Optional: Check Your Installation****

Run the following to verify the installation:

minikube version

kubectl version --client

### Troubleshooting

1. **If Minikube fails to start**:
   * Ensure your hypervisor (Hyper-V/Docker/VirtualBox) is installed and running.
   * Check the Minikube logs:
   * minikube logs
2. **Updating Minikube**:
3. minikube update-check
4. minikube update