==========
Docker Compose
==========

=> Docker Compose is a tool which is used to manage multi container based applications

=> Using Docker Compose we can easily setup & deploy multi container based applications

=> We will give containers information to Docker Compose using YML file (docker-compose.yml)

=> Docker Compose YML should have all the information related to containers creation

==========
Monolith Vs Microservices
==========

-> Monolith means single application will be available for all the functionalities

-> Microservices means collection apis will be available in the project / application

1) Products_Api
2) Cart_Api
3) Payment_Api
4) Orders_Api
5) Tracking_Api
6) Cancel_Api
7) Admin_Api
8) Reports_Api
9) Usermanament_api

=> Currently in the market we are developing Microservices Based applications

=> Microservices means collection apis will be available in the project / application

=> Every API should run in a seperate container

=> Running Multiple containers manully for all the apis is difficult job

********** To solve this problem Docker-Compose came into picture
****

=> Docker Compose is a tool which is used to manage multi container based applications

=> Using Docker Compose we can easily setup & deploy multi container based applications

=> We will give containers information to Docker Compose using YML file (docker-compose.yml)

=> Docker Compose YML should have all the information related to containers creation

==========
Docker Compose YML File
==========

version:

services:

network:

```
volumes:

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

=> Docker Compose default file name is  "docker-compose.yml"

# Create Containers using Docker Compose
$ docker-compose up

# Create Containers using Docker Compose with custom file name
$ docker-compose -f <filename> up

# Display Containers created by Docker Compose
$ docker-compose ps

# Display docker compose images
$ docker-compose images

# Stop & remove the containers created by docker compose
$ docker-compose down



====================
Docker Compose Setup
====================

# download docker compose
$ sudo curl -L "https://github.com/docker/compose/releases/download/1.24.0/docker-compose-$(uname -
s)-$(uname -m)" -o /usr/local/bin/docker-compose

# Give permission
$ sudo chmod +x /usr/local/bin/docker-compose

# How to check docker compose is installed or not
$ docker-compose --version



==========================================
Spring Boot with MySQL using Docker Compose
==========================================

=> Clone Git Repo : https://github.com/ashokitschool/spring-boot-mysql-docker-compose.git

=> Below is the docker compose file to dockerize spring boot application

---
version: "3"
services:
  application:
    image: springboot-app
    networks:
      - springboot-db-net
    ports:
      - "8080:8080"
    depends_on:
      - mysqldb
  mysqldb:
    image: mysql:5.7
    networks:
      - springboot-db-net
    environment:
      - MYSQL_ROOT_PASSWORD: root
      - MYSQL_DATABASE: sbms
```

```
networks:
        springboot-db-net:
...
```

=> Creat docker containers using above docker-compose file

```
$ docker-compose up -d

$ docker-compose ps

$ docker logs -f <container-name>

$ docker-compose down
```