# Docker Compose

# Agenda

**In this session, you will learn about:**

- Docker Compose Overview
- Docker Compose Features
- Docker Compose Use-Cases
- Docker Compose Example
- Understanding docker-compose file

# Docker Compose Overview

- Compose is a tool for defining and running multi-container Docker applications.
- Uses YAML file to configure application’s services.
- Single command creates and starts all the services from the configuration.
- Compose works in all environments: production, staging, development, testing, as well as CI workflows.
- Using Compose is basically a three-step process:
  - Define your app’s environment with a **Dockerfile** so it can be reproduced anywhere.
  - Define the services that make up your app in **docker-compose.yml** so they can be run together in an isolated environment.
  - Run **docker-compose up** and Compose starts and runs your entire app.

# Docker Compose Overview

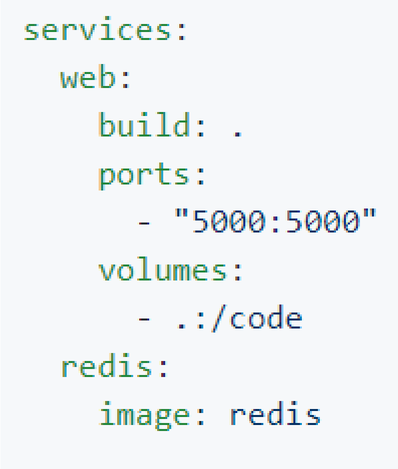
# Features

- Multiple isolated environments on a single host
- Preserve volume data when containers are created
- Only recreate containers that have changed
- Variables and moving a composition between environments

# Common use cases

- Development environments
- Automated testing environments
- Single host deployments

# Example - docker-compose.yml



```yaml
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - .:/code
  redis:
    image: redis
```

# Useful Commands

- **docker-compose up**
- **docker-compose ps**
- **docker-compose logs**
- **docker-compose scale**

# Lab- Docker Compose