Docker Compose

# Introduction to Docker Compose:

Compose is a tool for defining and running complex applications with Docker. With Compose, you define a multi-container application in a single file, then spin your application up in a single command which does everything that needs to be done to get it running. Compose uses YAML for its configuration file format.

# When to use Docker Compose:

Below, we’ll look at a couple of common scenarios that are made easier with Docker Compose.

## Development Environments

When developing services, it’s critical to have an isolated environment to interact with. You can use the Docker Compose command line tool to create and interact with your development environment.

First, you’ll need to add the configuration details for your applications’ service dependencies (databases, caches, web services) in a Compose YAML file. Then, you’ll be able to create and start one or more containers for each dependency with a single **docker-compose up** command. This helps simplify environment creation for other developers working on your team.

## Automate Testing

Compose can help provide you with the testing environments needed in a Continuous Integration/Deployment (CI/CD) workflow. Define the full testing environment, build it, run your tests, and destroy it in just a few commands.

# Installing Docker Compose:

Installing docker doesn't mean that you've installed docker-compose. It has as prerequisites that you've already installed the docker engine which you've already done. If you run Docker on OS X or Windows, Compose comes bundled with the Docker Toolbox. You should already have Docker Compose installed.

## Installing Compose on CentOS7:

Here are the commands you need to execute

$ curl -L https://github.com/docker/compose/releases/download/1.16.1/docker-compose-`uname -s`-`uname -m` > ./docker-compose

$ sudo mv ./docker-compose /usr/bin/docker-compose

$ sudo chmod +x /usr/bin/docker-compose

# Working with Multiple Containers using Docker Compose:

## Docker Linking Multiple Containers: Without Compose

### Example: Linking Wordpress and Mysql

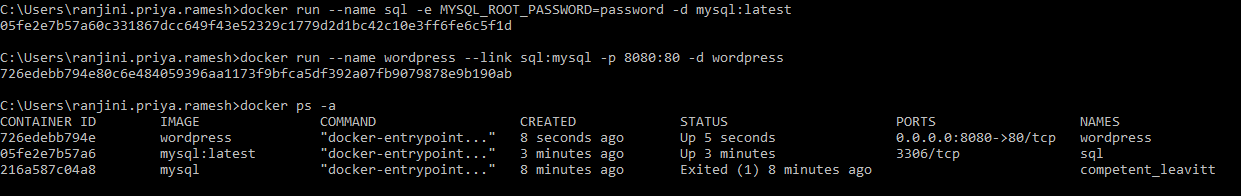
#### Pull mysql image:

docker pull mysql

#### pull wordpress image:

docker pull wordpress

#### Run mysql and wordpress:



### Running Mysql and Wordpress with arguments:

--name 🡪 user defined name for the image

-e 🡪 used to set environment variable

-d 🡪 to run as daemon

Latest 🡪 version of mysql to be run

--link 🡪 to link one image with another

docker run --name sql -e MYSQL\_ROOT\_PASSWORD=password -d mysql:latest

docker run --name wordpress --link sql:mysql -p 8080:80 -d wordpress

## Co-ordinate Multiple Containers using Compose:

### **Docker Compose Commands:**

docker-compose ps

docker-compose up

docker-compose start

docker-compose stop

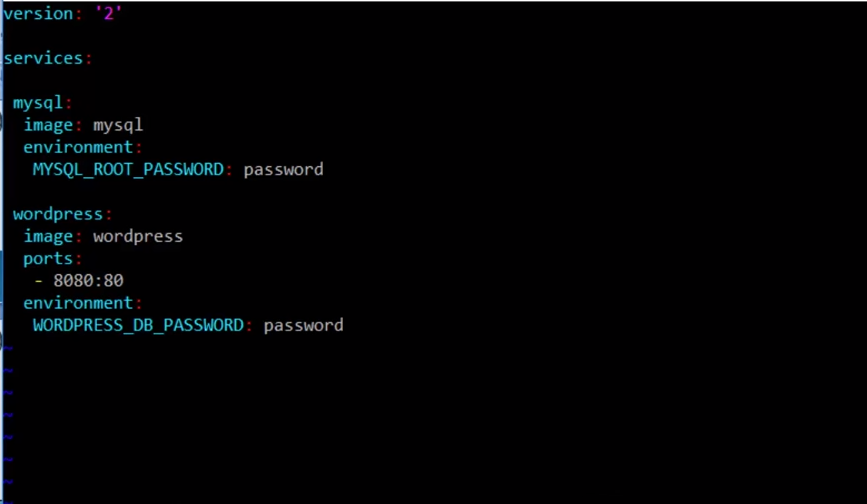
docker-compose restart

### **Compose Configuration file:**

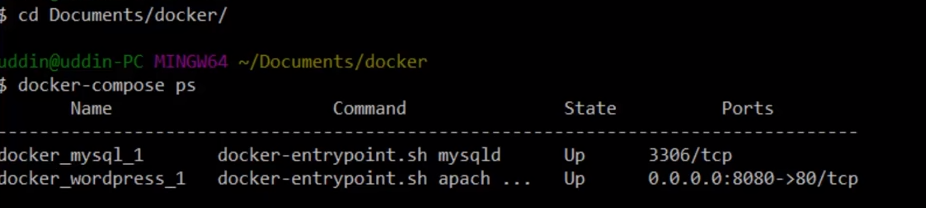


### **Steps:**

* Navigate to a directory where we would want to create a YML file
* Create yml file using touch docker-compose.yml command
* Edit the created docker-compose.yml file



* Run the application using docker-compose up command. Should run this command navigating to the directory path where the yml file exists. At some point wordpress and Mysql will be up.
* Navigate to the directory. Check for the list of containers using docker-compose ps



* This shows that both appears to be up and running.
* To stop use docker-compose stop.