# Docker - Compose

**Docker Compose** is used to run multiple containers as a single service. For example, suppose you had an application which required NGNIX and MySQL, you could create one file which would start both the containers as a service without the need to start each one separately.

In this chapter, we will see how to get started with Docker Compose. Then, we will look at how to get a simple service with MySQL and NGNIX up and running using Docker Compose.

## Docker Compose ─ Installation

The following steps need to be followed to get Docker Compose up and running.

**Step 1** − Download the necessary files from **github** using the following command −

curl -L "https://github.com/docker/compose/releases/download/1.10.0-rc2/dockercompose

-$(uname -s) -$(uname -m)" -o /home/demo/docker-compose

The above command will download the latest version of Docker Compose which at the time of writing this article is **1.10.0-rc2**. It will then store it in the directory **/home/demo/**.



**Step 2** − Next, we need to provide **execute privileges** to the downloaded Docker Compose file, using the following command −

chmod +x /home/demo/docker-compose



We can then use the following command to see the **compose** version.

### Syntax

docker-compose version

### Parameters

**version** − This is used to specify that we want the details of the version of **Docker Compose**.

### Output

The version details of Docker Compose will be displayed.

### Example

The following example shows how to get the **docker-compose** version.

sudo ./docker-compose -version

### Output

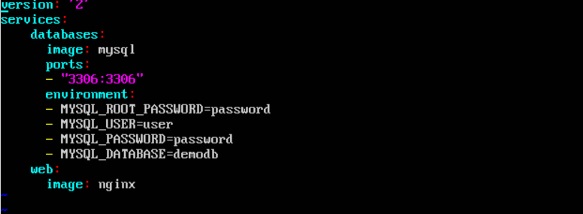
You will then get the following output −



## Creating Your First Docker-Compose File

Now let’s go ahead and create our first Docker Compose file. All Docker Compose files are YAML files. You can create one using the vim editor. So execute the following command to create the **compose** file −

sudo vim docker-compose.yml



Let’s take a close look at the various details of this file −

The **database** and **web** keyword are used to define two separate services. One will be running our **mysql** database and the other will be our **nginx** web server.

The **image** keyword is used to specify the image from **dockerhub** for our **mysql** and **nginx** containers

For the database, we are using the ports keyword to mention the ports that need to be exposed for **mysql**.

And then, we also specify the environment variables for **mysql** which are required to run **mysql**.

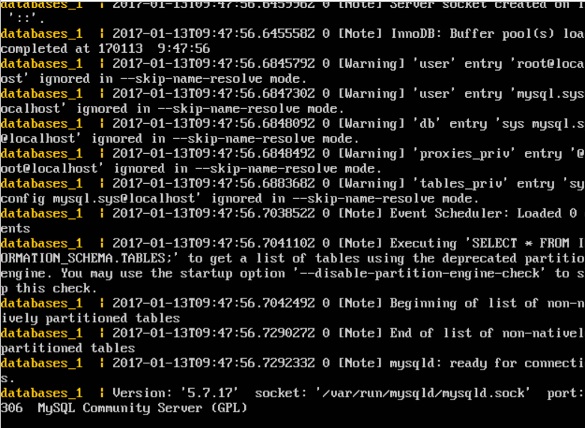
Now let’s run our Docker Compose file using the following command −

sudo ./docker-compose up

This command will take the **docker-compose.yml** file in your local directory and start building the containers.



Once executed, all the images will start downloading and the containers will start automatically.



And when you do a **docker ps**, you can see that the containers are indeed up and running.

