Docker-compose in action

Introduction to Docker Compose

We'll cover the following

In the last lesson, we successfully added the database to our app.

> You can clone the project using `git clone https://github.com/venky8283/Docker.git`

> Type `git checkout 4de325cf1da2428e757be4f2bcc53f35c384c598` to get to the code used in this lesson.

Summary of steps to add a database

1. Pull a Docker image of MySQL server
2. Create a SQL init script for the database
3. Run the database container with environment variables
4. Mount the init.sql script in the docker-entrypoint-init.db folder
5. Modify the code and rerun app container with a link to the database container

Every time you want to rerun the app, you have to go through these steps.
Wouldn’t it be great if we could automate these steps?

> The good news is docker-compose addresses this issue and automates these

> steps. In short, Compose is a tool for defining and running multi-container Docker applications.

Docker-compose installation

Follow the steps below to install docker-compose or check out the official installation for all platforms.

Windows and Mac

If you have installed Docker desktop, then docker-compose already comes with it. You can verify it using `docker-compose --version`.

Linux

Type the following commands in the terminal step by step:

- Run this command to download the current stable release of Docker Compose `sudo curl -L "https://github.com/docker/compose/releases/download/1.25.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose`
- Apply executable permissions to the binary with `sudo chmod +x /usr/local/bin/docker-compose`
- Verify installation with `docker-compose --version`.

Docker-compose file

Let's create a docker-compose file and see if we can automate this.

> If you checkout to commit "Docker-compose file for the app", you will see a docker-compose file there.

Modify mounts according to your files system. For example,

```
volumes:
 - "/usercode/:/code"
```

Here, replace 'usercode' directory to your directory where the project is cloned. It can be something like /Users/Home/username/documents/Docker.

Then, open up the terminal and cd into the directory where docker-compose.yml file is present.

Finally, hit `docker-compose up --build`.

You can also try running the code below. We will look at the commands in detail in the next lesson.

After a couple of minutes and a lot of steps, you will see that your app is running at 0.0.0.0:5000 with a database attached to it.

```
CREATE TABLE IF NOT EXISTS `users` (
  `user_id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) DEFAULT NULL,
  `password` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`user_id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=10001 ;

INSERT INTO `users` (`user_id`, `username`,  `password`) VALUES
(1, 'admin', 'admin123');
```

This is the power of docker-compose. In just one statement, you have your whole project set up and running.

I hope you are finding it interesting. Next, we will deep dive into docker-compose and understand each line in the docker-compose.yml file in the next lesson.