How to Install and use Docker Compose on Ubuntu 22.04

Installing Docker Compose

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a Compose file to configure your application's services. Then, using a single command, you create and start all the services from your configuration.
Use `curl` to download the Compose file into the `/usr/local/bin` directory.

```
curl -L
"https://github.com/docker/compose/releases/download/v2.12.2/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Output:

```
root@crown:~# curl -L
"https://github.com/docker/compose/releases/download/v2.12.2/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 664 100 664 0 0 2280 0 --:--:-- --:--:-- --:--:-- 2289
100 12.1M 100 12.1M 0 0 12.1M 0 0:00:01 0:00:01 --:--:-- 41.1M
```

Next, set the correct permissions so that the `docker-compose` command is executable.

```
chmod +x /usr/local/bin/docker-compose
```

To verify that the installation was successful, you can run the following command.

```
docker-compose --version
```

Output:

```
root@crown:~# docker-compose --version
Docker Compose version v2.12.2
```

(Optionally) Installing Docker on Ubuntu

Update the System.

```
apt update

apt upgrade
```

Install basic dependencies.

```
apt install apt-transport-https ca-certificates curl gnupg-agent
software-properties-common
```

Import docker repository GPG key.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-
key add -
```

Output:

```
root@crown:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg
| sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d
instead (see apt-key(8)).
OK
```

Add Docker CE repository.

```
add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

Output:

```
root@crown:~# add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Repository: 'deb [arch=amd64]
https://download.docker.com/linux/ubuntu jammy stable'
Description:
Archive for codename: jammy components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Hit:1 http://nl.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://nl.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://nl.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://nl.archive.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease
Reading package lists... Done
```

Installing Docker CE using the below command,

```
apt install docker-ce
```

To check status.

```
systemctl status docker
```

Output:

```
root@crown:~# systemctl status docker
● docker.service - Docker Application Container Engine
Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor
preset>
Active: active (running) since Wed 2022-04-06 20:18:28 UTC; 25min ago
TriggeredBy: ● docker.socket
Docs: https://docs.docker.com
Main PID: 42908 (dockerd)
Tasks: 12
Memory: 87.8M
CPU: 40.617s
CGroup: /system.slice/docker.service
└─42908 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con>
```

(Optionally) Setting Up a docker-compose.yml file

In this section, we'll use Docker Compose to construct a multi-container WordPress utility.
creating a new directory in your home folder, and then moving into it.

```
mkdir my_app

cd my_app
```

Next, create the `docker-compose.yml` file.

```
nano docker-compose.yml
```

Paste the following content on your `docker-compose.yml` file.

```
version: '3'

services:
db:
image: mysql:5.7
restart: always
volumes:
- db_data:/var/lib/mysql
environment:
```

```
MYSQL_ROOT_PASSWORD: password
MYSQL_DATABASE: wordpress

wordpress:
image: wordpress
restart: always
volumes:
- ./wp_data:/var/www/html
ports:
- "8080:80"
environment:
WORDPRESS_DB_HOST: db:3306
WORDPRESS_DB_NAME: wordpress
WORDPRESS_DB_USER: root
WORDPRESS_DB_PASSWORD: password
depends_on:
- db

volumes:
db_data:
wp_data:
```

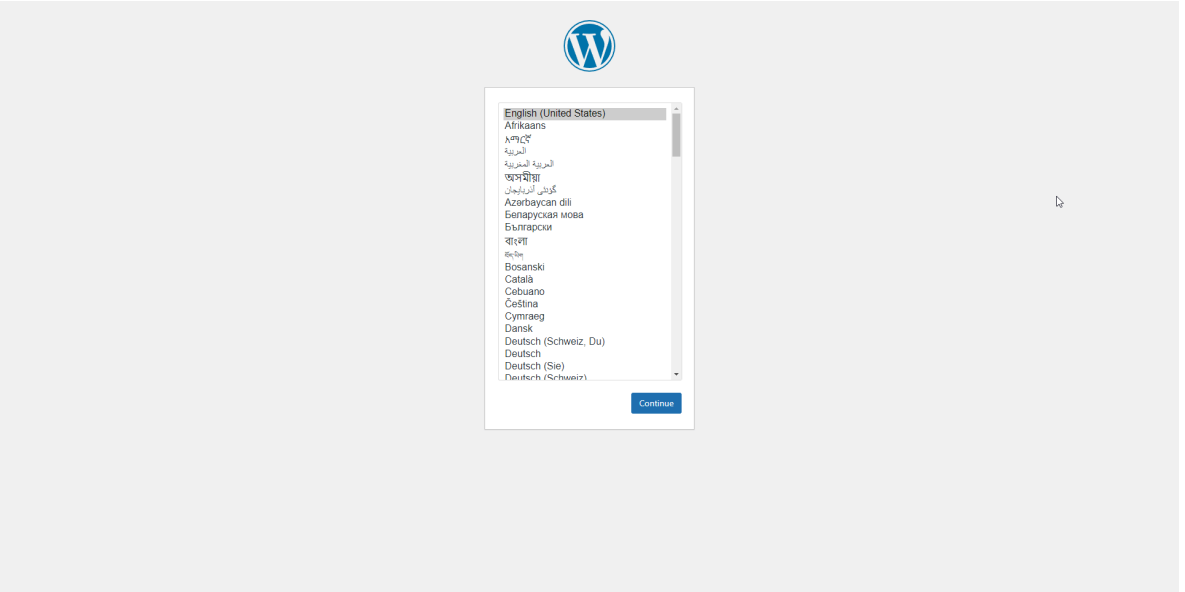
In this example, we have services, db, and wordpress. Each service runs one image, and creates a separate container when docker-compose is run.
Start up the WordPress application by running the following command.

```
docker-compose up
```

Output:

```
root@crown:~# docker-compose up
Creating network "root_default" with the default driver
Creating volume "root_db_data" with default driver
Creating volume "root_wp_data" with default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
f7ec5a41d630: Pull complete
9444bb562699: Pull complete
6a4207b96940: Pull complete
181cefd361ce: Pull complete
```

Navigate to your browser. http://yourserver-ip-address:8080 and you will see the WordPress installation screen.



Start the Compose in a detached mode by the following command.

```
docker-compose up -d
```

Output:

```
root@crown:~# docker-compose up -d
Starting root_db_1 ... done
Starting root_wordpress_1 ... done
```

To check the running services.

```
docker-compose ps
```

Output:

```
root@crown:~# docker-compose ps
Name Command State Ports
------------------------------------------------------------------------
-----------
root_db_1 docker-entrypoint.sh mysqld Up 3306/tcp, 33060/tcp
root_wordpress_1 docker-entrypoint.sh apach ... Up 0.0.0.0:8080-
>80/tcp
```

To stop the services only.

```
docker-compose stop
```

To stop and remove containers and networks.

```
docker-compose down
```

This concludes the Installation and usability of Docker Compose on Ubuntu 22.04.