Install Docker Compose on Linux Operating Systems

computingforgeeks.com/how-to-install-latest-docker-compose-on-linux

Josphat Mutai

July 20, 2024

Docker Compose is an open source tool created for running multi container-based Docker applications. With Docker Compose, you easily a deployment with multiple container applications in a single file named `docker-compose.yml`. Then use the file definitions to spin up and manage your full application stack with single command `docker-compose`. Below is a simple docker-compose.yml file that can be used to create a web application with web server and database containers.

```
$ vim docker-compose.yml
services:
  web:
    image: nginx:latest
    ports:
      - "8080:80"
  db:
    image: postgres:latest
    environment:
      POSTGRES_PASSWORD: DBPassword
```

This post aims to be a concise instructional step-by-step guide for developers and SysAdmins seeking to setup Docker Compose on Linux. We will check the Github API releases page for the project, and pull the latest binary file.

How To Install Docker Compose on Linux

Follow the steps below to install and use Docker Compose on Linux from binary file.

1. Install Docker Compose

You need curl and wget installed on your system for this operation. And definitely, access to the Terminal as a user with sudo privileges.

```
### CentOS / RHEL ###
sudo yum -y install curl wget

### Debian / Ubuntu ###
sudo apt update
sudo apt install -y curl wget

### Fedora ###
sudo dnf -y install curl wget
```

Once curl has been installed, download the latest Compose on your Linux machine.

```
curl -s https://api.github.com/repos/docker/compose/releases/latest | grep
browser_download_url  | grep docker-compose-linux-x86_64 | cut -d '"' -f 4 | wget
-qi -
```

Make the binary file executable.

```
chmod +x docker-compose-linux-x86_64
```

Move the file to your PATH.

```
sudo mv docker-compose-linux-x86_64 /usr/local/bin/docker-compose
```

Confirm version.

```
$ docker-compose version
Docker Compose version v2.28.1
```

Add user to docker group:

```
sudo usermod -aG docker $USER
newgrp docker
```

2. Configure docker-compose shell completion

Compose has command completion for the bash and zsh shell.

For Bash users

Place the completion script in `/etc/bash_completion.d`/.

```
sudo mkdir -p /etc/bash_completion.d/
sudo curl -L
https://raw.githubusercontent.com/docker/compose/master/contrib/completion/bash/docker-compose -o /etc/bash_completion.d/docker-compose
```

Source the file or re-login to enjoy completion feature.

```
echo "source /etc/bash_completion.d/docker-compose"|tee -a ~/.bashrc
source /etc/bash_completion.d/docker-compose
```

For Zsh users

Download the completion script in your `~/.zsh/completion/`

```
mkdir -p ~/.zsh/completion
curl -L
https://raw.githubusercontent.com/docker/compose/master/contrib/completion/zsh/_docker-compose > ~/.zsh/completion/_docker-compose
```

Include the directory in your `$fpath` by adding in `~/.zshrc`:

```
$ vim ~/.zshrc||nano ~/.zshrc
fpath=(~/.zsh/completion $fpath)
```

Make sure `compinit` is loaded or do it by adding in `~/.zshrc`:

```
autoload -Uz compinit && compinit -i
```

Then reload your shell:

```
exec $SHELL -l
```

3. Test Docker Compose installation

Our comprehensive guide is on Managing Docker Containers with Docker Compose

Create a test Docker Compose file.

```
vim docker-compose.yml
```

Add below data to the file.

```
services:
  web:
    image: nginx:latest
    ports:
     - "8080:80"
    links:
     - php
  php:
    image: php:8-fpm
```

Start service containers.

```
$ docker-compose up -d
[+] Running 18/18
 ✔ php Pulled
20.9s
   ✔ 01c187ab622c Pull complete
1.2s
   ✔ 4382a8829fff Pull complete
15.3s
   ✔ 43046b340e34 Pull complete
15.3s
   ✔ 41722365abab Pull complete
15.5s
   ✔ a52941633aa9 Pull complete
15.5s
   ✔ 930f8db3b95e Pull complete
18.9s
   ✔ f32aed4faf2d Pull complete
18.9s
   ✔ 499f39c692f7 Pull complete
18.9s
   ✔ add8a6605e0d Pull complete
19.0s
 ✔ web Pulled
6.6s
   ✔ 2cc3ae149d28 Already exists
0.0s
   ✔ 1018f2b8dba8 Pull complete
4.6s
   ✔ b831e78d8e20 Pull complete
4.7s
   ✔ 3ab22521e919 Pull complete
4.7s
   ✔ 5112bf42775b Pull complete
4.7s
   ✔ cbdaf9e4ee2d Pull complete
4.7s
   ✔ a06b6fd631e8 Pull complete
4.7s
[+] Running 3/3
 ✔ Network root_default  Created
0.1s
 ✔ Container root-php-1  Started
0.7s
 ✔ Container root-web-1  Started
0.8s
```

Show running Containers

```
$ docker-compose ps
   Name                 Command               State                  Ports
----------------------------------------------------------------------------------
--------
root_php_1   docker-php-entrypoint php-fpm    Up      9000/tcp
root_web_1   /docker-entrypoint.sh ngin ...   Up      0.0.0.0:8080-
>80/tcp,:::8080->80/tcp
```

Destroy containers

```
$ docker-compose stop
Stopping root_web_1 ... done
Stopping root_php_1 ... done

$ docker-compose rm -f
Going to remove root_web_1, root_php_1
Removing root_web_1 ... done
Removing root_php_1 ... done
```

Go through Official Docker documentation and Docker Compose documentation to learn more.

More guides:

- Ctop – Top command for container metrics
- How to Install Portainer Docker UI manager
- How To run Local Kubernetes clusters in Docker
- Deploy Lightweight Kubernetes with MicroK8s and Snap
- How to run Minikube on KVM