Install Docker on Linux

No matter your distribution of choice, you'll need a 64-bit installation and a kernel at 3.10 or newer. Kernels older than 3.10 do not have the necessary features Docker requires to run containers; data loss and kernel panics occur frequently under certain conditions.

Check your current Linux version with `uname -r`. You should see something like `3.10.[alphanumeric string].x86_64`.

Debian and Ubuntu

Docker runs on:

- Ubuntu Xenial 16.04 LTS
- Ubuntu Wily 15.10
- Ubuntu Trusty 14.04 LTS
- Ubuntu Precise 12.04 LTS
- Debian testing stretch
- Debian 8.0 Jessie
- Debian 7.0 Wheezy (you must enable backports)

Debian Wheezy

If so, you need to enable backports (if not, ignore this section):

1. Log into the system and open a terminal with `sudo` or `root` privileges (or run `sudo -i` from your terminal).
2. Open `/etc/apt/sources.list.d/backports.list` with your favorite text editor (if the file does not exist, create it).
3. Remove existing entries.
4. Add an entry for backports on Debian Wheezy:

```
deb http://http.debian.net/debian wheezy-backports main
```

5. Update your packages:

```
apt-get update -y
```

Ubuntu Precise 12.04

If so, you need to make sure you have the 3.13 kernel version. You must upgrade your kernel:

1. Open a terminal on your system.
2. Update aptitude:

```
sudo apt-get update -y
```

3. Install the additional packages:

```
sudo apt-get install -y linux-image-generic-lts-trusty linux-headers-gene
```

4. On a graphical Ubuntu environment, you need to additionally run the following:

```
sudo apt-get install -y xserver-xorg-lts-trusty libgl1-mesa-glx-lts-trust
```

5. Reboot your system:

```
sudo reboot
```

Update Aptitude

1. Log onto your system with a user with `sudo` privileges.
2. Open a terminal window.
3. Purge the older repositories:

```
sudo apt-get purge -y lxc-docker* && sudo apt-get -y purge docker.io*
```

4. Update your packages, making sure `apt` works with `https` and the server has CA certificates:

```
sudo apt-get update -y && sudo apt-get install -y apt-transport-https ca-
```

5. Get the new GPG key:

```
sudo apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-
```

6. Open or create the file `/etc/apt/sources.list.d/docker.list` in your favorite text editor (you need `sudo` or `root` for this).
7. Add an entry for your OS

Version	Source
Ubuntu Precise 12.04 LTS	`deb https://apt.dockerproject.org/repo ubuntu-precise main`
Ubuntu Trusty 14.04 LTS	`deb https://apt.dockerproject.org/repo ubuntu-trusty main`
Ubuntu Wily 15.10 LTS	`deb https://apt.dockerproject.org/repo ubuntu-wily main`
Ubuntu Xenial 16.04 LTS	`deb https://apt.dockerproject.org/repo ubuntu-xenial main`
Debian Wheezy	`deb https://apt.dockerproject.org/repo debian-wheezy main`
Debian Jessie	`deb https://apt.dockerproject.org/repo debian-jessie main`
Debian Stretch/Sid	`deb https://apt.dockerproject.org/repo debian-stretch main`

8. Save and close the file.
9. Update Aptitude again:

```
sudo apt-get update -y
```

10. Verify Aptitude pulls from the right repository:

```
sudo apt-cache policy docker-engine
```

Install Docker

If you use Ubuntu Trusty, Wily, or Xenial, install the `linux-image-extra` kernel package:

```
sudo apt-get update -y && sudo apt-get install -y linux-image-extra-$(uname -
```

1. Install Docker:

   ```
   sudo apt-get install docker-engine -y
   ```

2. Start Docker:

   ```
   sudo service docker start
   ```

3. Verify Docker:

   ```
   sudo docker run hello-world
   ```

The Docker Group

If you prefer, you can set up a `docker` group to run Docker (instead of `root`). *However*, as `docker` must have `sudo` access, `docker` receives the same access as `root`.

1. Run the following command to create a Docker group on Ubuntu:

   ```
   sudo groupadd docker && sudo usermod -aG docker ubuntu
   ```

2. Log out and back in.

3. Run the following command to create a Docker group on Debian:

```
sudo groupadd docker && sudo gpasswd -a ${USER} docker && sudo service do
```

> *You may specify a user instead of `${USER}` if you prefer.*

4. Verify a successful Docker installation:

```
docker run hello-world
```

Red Hat Enterprise Linux (RHEL) and CentOS

Docker runs on RHEL 7 and CentOS 7.

Install Docker

Install with Yum

1. Log into your system as a user with `sudo` privileges.
2. Update your system: `sudo yum update -y`.
3. Add the yum repo (use the code below for both RHEL 7 *and* CentOS 7):

```
$ sudo tee /etc/yum.repos.d/docker.repo <<-'EOF'
[dockerrepo]
name=Docker Repository
baseurl=https://yum.dockerproject.org/repo/main/centos/7/
enabled=1
gpgcheck=1
gpgkey=https://yum.dockerproject.org/gpg
EOF
```

4. Install Docker:

```
sudo yum install docker-engine -y
```

5. Start Docker:

```
sudo service docker start
```

6. Verify Docker:

```
sudo docker run hello-world
```

Install with the Docker Installation Script

1. Log into your system as a user with `sudo` privileges.
2. Update your system:

```
sudo yum update -y
```

3. Run Docker's installation script:

```
curl -fsSL https://get.docker.com | sh;
```

> *This script adds the `docker.repo` repository and installs Docker.*

4. Start Docker:

```
sudo service docker start
```

5. Verify Docker:

```
sudo docker run hello-world
```

The Docker Group

If you prefer, you can set up a `docker` group to run Docker (instead of `root`). *However*, as `docker` must have `sudo` access, `docker` receives the same access as `root`.

1. Run the following command to create a Docker group and add your user to the group (replace USERNAME with your username):

```
sudo groupadd docker && sudo usermod -aG docker USERNAME
```

2. Log out and back in.
3. Verify Docker works without `sudo`:

```
docker run hello-world
```

Start Docker at Boot

Run one of the following:

- `sudo chkconfig docker on`
- `sudo systemctl enable docker`

Common Issues

Note: Members in the docker group have *root* privileges. Hardening Docker is covered in a future tutorial.

Ubuntu

Ubuntu Utopic 14.10 and 15.05 exist in Docker's `apt` repository without official support. Upgrade to 15.10 or [preferably] 16.04. If you use Ubuntu 12.04, you need to update your kernel.

Debian

If you run Debian Wheezy, you need to update the sources with backports.

"Cannot connect to the Docker daemon. Is 'docker daemon' running on this host?"

If you get this error, you need to unset DOCKER_HOST; run `unset DOCKER_HOST` to clear the variable.

Next: Using Docker Hub

All about how to use Docker's hosted registry.

About

Privacy Policy