

Installing Docker on Ubuntu

Install

- Run this:

```
### Everything below can be copy/pasted as a one-liner
. /etc/os-release && \
sudo apt update && \
sudo apt install -y apt-transport-https ca-certificates curl software-properties-
common && \
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add - && \
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$VERSION_CODENAME stable" && \
sudo apt update && \
sudo apt install -y docker-ce docker-compose && \

# Verify it's running
sudo systemctl status docker -n 0

# Update (create if need be) the docker configuration to use the CGNAT CIDR range for
container networking
sudo apt install -y moreutils jq && \
sudo bash -c 'if [ ! -f "/etc/docker/daemon.json" ] ; then echo "{}" >
/etc/docker/daemon.json ; fi ; jq '\''."default-address-pools"[0].base =
"100.64.0.0/16" | ."default-address-pools"[0].size = 24'\'' /etc/docker/daemon.json |
sponge /etc/docker/daemon.json' ; cat /etc/docker/daemon.json

# Restart docker to pull in changes
sudo systemctl restart docker

# Set up cron task to clean up stale, unused docker images
sudo bash -c 'cat << EOF > /etc/cron.daily/docker-image-prune
#!/bin/bash
### Prune images older than 7 days
```

```
docker image prune -af --filter "until=$((7*24))h" 2> /dev/null > /dev/null
EOF
'
sudo chmod u+x /etc/cron.daily/docker-image-prune
```

Per-admin first-time setup

For each local user you want to grant docker access (such as your own user)...

- Run this:

```
# Add yourself to the Docker group
sudo usermod -aG docker ${USER}

# Disconnect and reconnect to apply permissions OR start a new privileged shell...
sudo su - ${USER}
```

The above only has to be run once. After that, you have persistent access every time you log in.

Optional Install of Portainer

Gives users an approachable GUI to manage containers

- Install via docker-compose

```
sudo mkdir -p /opt/docker-compose/portainer
cd /opt/docker-compose/portainer
sudo vi docker-compose.yml
```

- Paste in the following and save/exit:

```
services:
  portainer:
    image: portainer/portainer-ce
    restart: always
    ### Generated with:
    #   docker run --rm httpd:2.4-alpine htpasswd -nbB admin 'YOUR_PASSWORD_HERE' |
    cut -d ":" -f 2 | sed -Ee 's/\$/\$\$/g'
```

```
command: --admin-password
'$$2y$$05$$_CHANGE_THIS_HASH_USING_THE_OUTPUT_FROM_THE_COMMAND_ABOVE'
ports:
  - "9000:9000"
  - "9443:9443"
  - "8000:8000"
volumes:
  - portainer_data:/data
  - /var/run/docker.sock:/var/run/docker.sock

volumes:
  portainer_data:
```

- Set the admin password

```
ADMIN_PW_HASH=$(docker run --rm httpd:2.4-alpine htpasswd -nbB admin "`read -p "Enter
your new admin password: " -s i ; echo $i`" | cut -d ":" -f 2 | sed -Ee
's/\$/\$/g') ; echo "" ; \
echo $ADMIN_PW_HASH | sed -Ee 's/\$/\$/g' | sed -Ee 's//\//g' | sudo sed -Ee
"s/--admin-password '[^']*'/--admin-password '`cat -`'/g" -i docker-compose.yml
```

- Run:

```
docker compose up -d && docker compose logs -f
```

You can stop the log output with Ctrl+C

- Open the UI via <https://your.hostname.or.ip:9443/>
- Bypass the invalid cert warning
- Login with the admin password you set above

Revision #12

Created 2024-07-09 18:13:06 UTC by jholmstadt

Updated 2025-12-16 19:30:49 UTC by jholmstadt