How to Install Docker on the Linux Server?

written by sysadmin | 10 February 2025

A Docker is a platform for developing, shipping, and running container applications. Docker is like installing a virtual machine application on your laptop or server, whether it’s VirtualBox, VMWare, or Xen, so you can test various operating systems or applications on it without putting your laptop or server in danger.

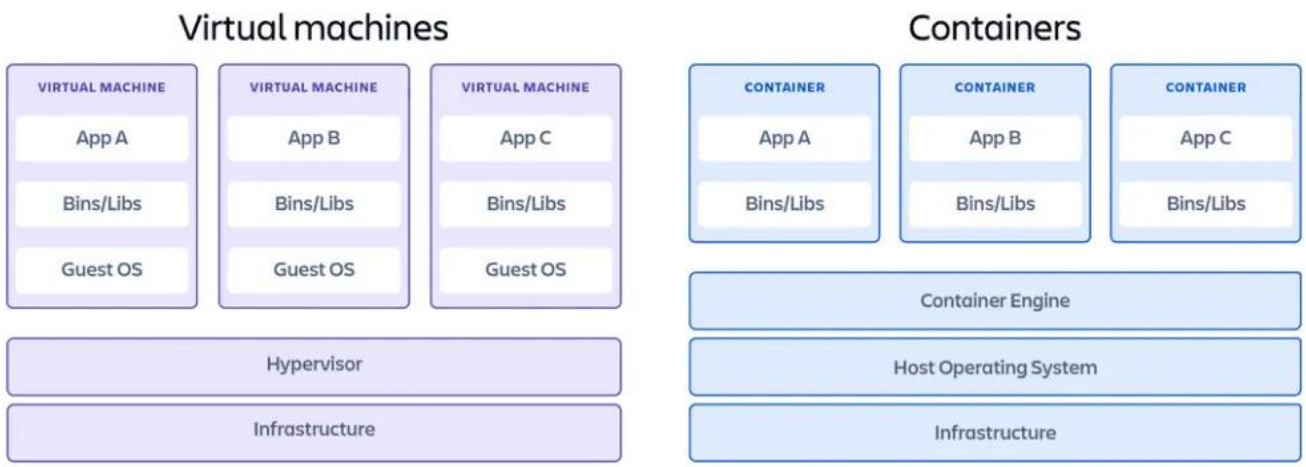
Problem

How to install Docker on the Linux server?

Solution

A. Docker summary

The key differentiator between containers and virtual machines is that virtual machines virtualize an entire machine down to the hardware layers, and containers only virtualize software layers above the operating system level. Take a look at the image below to make the difference between virtual machines and Docker clearer:



Comparison of Docker and Virtual Machine Architecture (image credit from atlassian.com)

The table below shows the comparison between virtual machines and Docker:

Comparison Item	Docker Container	VM
Isolation level	Low	High
Time required for startup	Seconds	Minutes
Image size	Several megabytes	Hundreds of megabytes to several gigabytes
Running performance (compared with bare metal servers)	Performance loss: < 2%	Performance loss: about 15%
Image portability	Not related to the platform	Related to the platform
Density	100 to 1000 on a single machine	10 to 100 on a single machine
Security	1. When the privilege of a user in a container is escalated from a common user to the **root** user, the user gains root permissions of the host machine. 2. Hardware isolation is not implemented, so containers are vulnerable to attacks.	1. The root permissions of a VM tenant are isolated from those of the host machine. 2. Hardware isolation is implemented to prevent VM escape and data exchange.

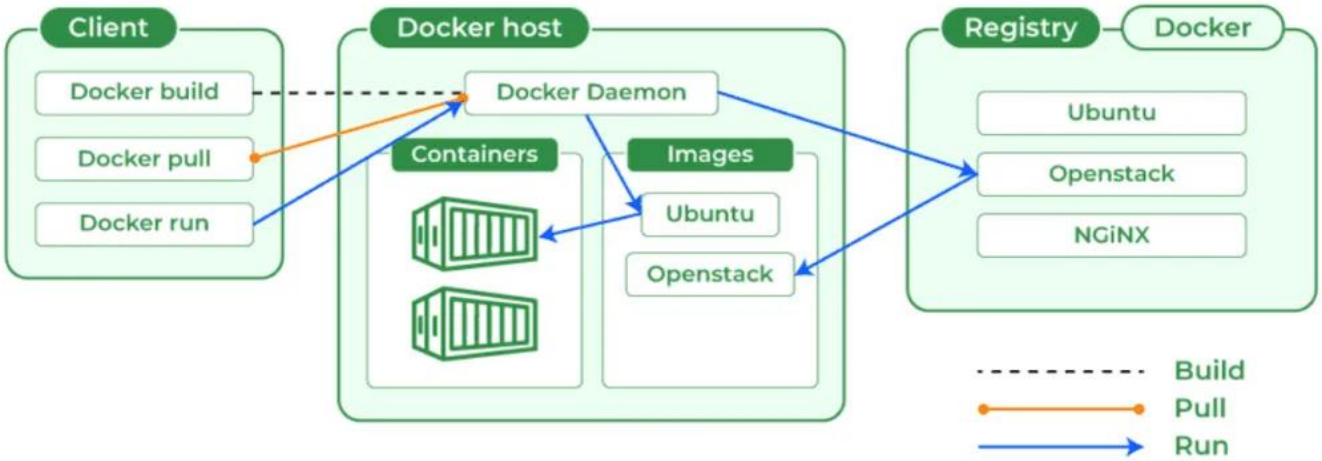
Comparison between Docker and virtual machine (Image credit from huawei.com)

Below is a brief explanation of the terms in Docker:

- □□□□□□ □□□□: Downloads images from Docker Hub if not locally available
- □□□□□□ □□□□□: Creates a local image using a Dockerfile, enabling custom images
- □□□□□□ □□□□:Uploads images to Docker Hub, allowing sharing
- □□□□□□ □□□: Takes an image to run a container, useful for starting web servers or other applications
- □□□□□□ □□□□□: Read-only templates forming the base of containers, including all application dependencies
- □□□□□□ □□□□□□: Interacts with Docker, sending instructions to the Docker daemon to execute tasks
- □□□□□□ **Daemon**: Handles all requests, including building, running, and distributing containers
- □□□□□□ □□□□□□□□□□: Include everything needed to run an application, such as code, libraries, and configurations

- □□□□□□ □□□□□□□□: Stores Docker images, with Docker Hub as a public registry and the option to create private ones

The image below is a picture of how the Docker works:



How Docker works (Image credit from geeksforgeeks.org)

B. Install Docker

In general, use the command below to install Docker on Linux:

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

But after you execute the commands above, there is an error like this when you install it in RockyLinux:

```
ERROR: Unsupported distribution ‘rocky’
```

You have to install Docker manually using these commands:

```
sudo dnf config-manager --add-repo
https://download.docker.com/linux/rhel/docker-ce.repo
sudo dnf -y install docker-ce docker-ce-cli containerd.io docker-buildx-
plugin docker-compose-plugin
```

Or when you install Docker in OpenSUSE, you get an error like this:

ERROR: Unsupported distribution ‘opensuse-leap’

Use the commands below to install Docker in OpenSUSE:

```
sudo zypper install -y docker docker-compose docker-compose-switch
```

C. After installing Docker

Use the following command to run Docker:

```
sudo systemctl restart docker
sudo systemctl enable docker
```

To see the version of Docker you installed, use the command below:

```
docker info
```

```
sysadmin@ubuntu2404:~$ docker info
Client: Docker Engine - Community
 Version:    27.5.1
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version:  v0.20.0
    Path:     /usr/libexec/docker/cli-plugins/docker-buildx
  compose: Docker Compose (Docker Inc.)
    Version:  v2.32.4
    Path:     /usr/libexec/docker/cli-plugins/docker-compose

Server:
ERROR: permission denied while trying to connect to the Docker daemon socket at u
nix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.47/info": d
ial unix /var/run/docker.sock: connect: permission denied
errors pretty printing info
sysadmin@ubuntu2404:~$
```

Running the docker info command

D. Test the application in Docker

After that, to see whether Docker is running well on the server, use the command below to run the hello-world

container on your server:

```
docker run hello-world
```

If you have got the error like the image below:

```
sysadmin@ubuntu2404:~$ docker run hello-world
docker: permission denied while trying to connect to the Docker daemon socket at
unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial
 unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
sysadmin@ubuntu2404:~$
```

Error when running the docker run command

ERROR: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get “http://%2Fvar%2Frun%2Fdocker.sock/v1.47/info”: dial unix /var/run/docker.sock: connect: permission denied

Then you have to run the following command:

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

Log out of your server and log in again. After that, you should be able to run the Docker commands like in the image below:

```
sysadmin@ubuntu2404:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:d715f14f9eca81473d9112df50457893aa4d099adeb4729f679006bf5ea12407
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

sysadmin@ubuntu2404:~$
```

Test the Docker run command

That way, your Docker application is ready to use.

Note

You don’t have to use Docker to create and run containers, but you can use other applications such as podman, buildah, cri-o, etc. However, Docker is the most popular at the moment. Also, the terms and workings of various container applications are almost the same, so if you understand the terms and workings of Docker, then you will also understand the terms and workings of other container applications. To learn about basic commands in Docker, go to this page.

References

phoenixnap.com
youtube.com
docs.docker.com
geeksforgeeks.org
atlassian.com
qa.com
info.support.huawei.com
simform.com
linkedin.com
docs.rockylinux.org
en.opensuse.org