

How to Run Docker on a VM

Running Docker on a VM isn't tricky. Docker is nothing more than a standard application or service. In this article, we'll create a virtualized Linux environment and then install the Docker app.

To complete this process, you should understand how to work with the command-line environment in Linux and the package manager for your Linux distribution of choice. We'll assume you're targeting and using Ubuntu for the remaining portion of this article.

But first, you'll need to create a virtual machine using the Ubuntu OS. There are many ways to make virtual machines, but that's beyond the scope of this article. Consult with the IT manager in your organization to discuss the specifics of your organization's virtualization environment.

After your Ubuntu VM is operational, go ahead and SSH into that VM. Now, let's install Docker.

How to Install and Run Docker on a VM

1. First, we need to update the Apt package manager to ensure that you have the latest packages indexed in the Apt repositories on your computer. In the terminal, run the command below:

```
sudo apt update
```

2. After Apt updates its package index, we need to ensure that Apt can use SSL connections to download packages from its index. To do that, enter the command below:

```
sudo apt install ca-certificates curl gnupg lsb-release
```

3. Give Apt a minute or two to install those packages. After that, it may ask for verification. If Apt does prompt for confirmation, review the changes in the command-line, and if you're satisfied with those changes, accept them.

4. After Apt is done updating, we need to add the official Docker repository to Apt's index. Enter the command below to add the Docker repo to your Apt index:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo  
gpg --dearmor -o /usr/share/keyrings/docker-archive-  
keyring.gpg
```

5. We're close to being done. Now that Apt knows where to find the official Docker repositories, we need to tell Apt to only use the Stable branch for Docker packages. Enter the command below for that:

```
echo \"deb [arch=$(dpkg --print-architecture) signed-  
by=/usr/share/keyrings/docker-archive-keyring.gpg]  
https://download.docker.com/linux/ubuntu \ $(lsb_release -cs)  
stable\" | sudo tee /etc/apt/sources.list.d/docker.list >  
/dev/null
```

6. Finally, let's update the Apt index again so Apt knows where to find the Docker packages and install Docker with the commands below:

```
sudo apt update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

New Features: As of 2024, Docker has introduced several enhancements such as [Docker Init for streamlined project setup](#), [Docker Scout for security vulnerability detection](#), and [Resource Saver mode](#) for minimizing system resource usage when idle. Additionally, [Docker Build Cloud](#) now allows faster builds through cloud infrastructure

That's it! Docker is now installed and operational within your Linux VM. Are you ready to test Docker?

How to Create a Private Cloud with Docker

One of the best reasons to use Docker is to create personalized and private cloud systems. Docker is an excellent tool for this. So, let's install OwnCloud.

1. In the Ubuntu terminal, enter the command below:

```
docker pull owncloud/server
```

2. Docker may take some time to download that image and start running it. After

Docker completes that process, open your browser and navigate to the IP address of your virtual machine running Docker at port 8080 (<https://192.168.1.20:8080>).

3. If all goes well, you should see an OwnCloud login screen.

Bonus: Can I Use Docker as a VM?

Though it's not widely recommended, you can use Docker to run virtual machines. That is, you can kind of use Docker to run a virtual machine. Let's take a closer look at this.

Docker isn't a virtual machine platform. It's a container platform. There's a distinct difference.

Docker doesn't create virtualized PC components like Hyper-V, VirtualBox, or ESXi. Instead, virtual machines create additional overhead by emulating PC hardware. Your virtual machines are essentially a bunch of minicomputers living inside of a larger computer.

Docker shares PC resources with its containers. It doesn't create virtualized PC hardware. The CPU that your host OS uses is the same CPU that your Docker containers use. It's not segmented for virtualization. To the Docker container, there's no difference in that CPU.

The hardware resources that Docker uses live in a sandbox. That means that though the host OS and the Docker container use the same CPU and same CPU cycles, the threads that the Docker container uses have a big fence around them. The same is true for memory and other system components.

Running another OS inside a Docker container isn't true virtualization. However, that doesn't mean you can't simulate a virtualized environment with Docker. For example, search the Docker Hub for "Xfce." You'll find containers that host full OS versions of Ubuntu with the Xfce desktop.

Security Considerations: As of 2024, [updated security practices](#) include avoiding exposure of the Docker daemon socket and selecting secure base images. Additionally, vulnerabilities like [CVE-2024-21626](#) have been addressed through regular updates.

Here's the catch. Docker is a Linux-only service. Docker can be run on Windows, but Windows uses WSL to use Docker. Otherwise, it's still running on top of Linux. Docker depends on the Linux kernel to operate correctly. Since Docker doesn't virtualize components, its host OS needs to have access to the Linux kernel and

other Linux components. That means you can only use Docker to run the Linux OS and not Windows or Mac OS. If you need to run Windows or OS X, you're stuck with virtualization.

Here's a quick set of instructions for running a Docker container with Ubuntu:

1. Open a terminal in Ubuntu.
2. Run this command:

```
docker run -it -p 9096:3389 -e 3389 --shm-size 2g  
manishfoodtechs/xfcefulldesktop_ubuntu20.4
```

3. In Windows, open RDP and connect to your local IP address with port 9906.
4. Use the username "root" with the password "123456" to log in.

In this article we walked you through the steps for creating a virtualized Linux environment and then installing the Docker app. Check out the documentation for more information on running Docker containers.