

Install Docker on Ubuntu 18.04

DISCLAIMER

Last updated April 10, 2021

INTRODUCTION

The information provided by site <https://tech2talk.github.io/vms-and-containers/> or <https://github.com/tech2talk/vms-and-containers> is for informational purposes only. All information on the aforementioned sites and within source code or within documents contained on them is provided “AS IS”. I make no representation or warranty of any kind, express or implied, regarding the accuracy, adequacy, validity, reliability, availability or completeness of any information contained or hosted within these sites.

Under no circumstance shall I have any liability to you or any user for any loss or damage of any kind incurred as a result of the use of or reliance on any information provided on the aforementioned sites or the documents contained within them. Your use of the aforementioned sites and your reliance on any information contained within documents including but not limited to source code is solely at your own risk. This [disclaimer template](#) was created using Termly.

EXTERNAL LINKS DISCLAIMER

The documents stored or hosted on the aforementioned sites contain links to other websites or content belonging to or originating from third parties or links to websites and features in banners or other advertising. Such external links are not investigated, monitored, or checked for accuracy, adequacy, validity, reliability, availability or completeness by me.

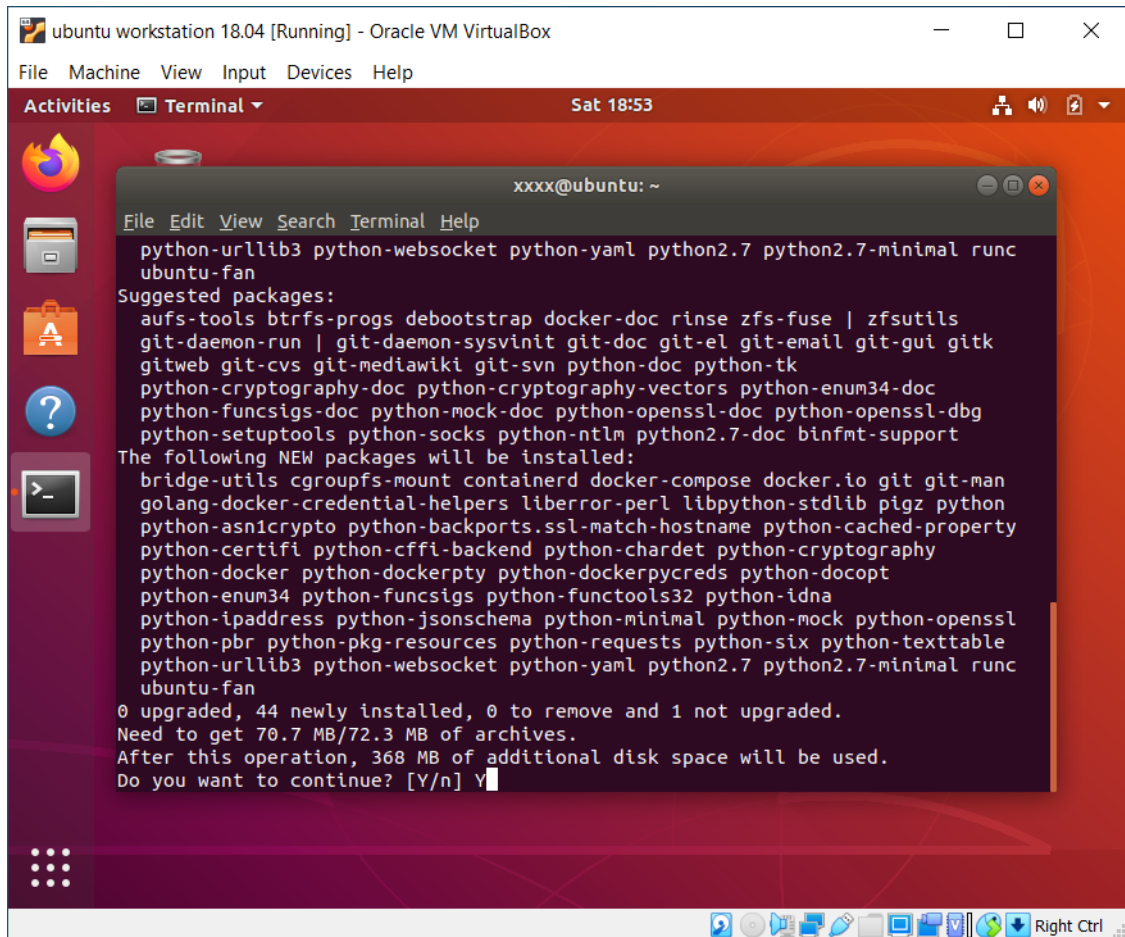
I do not warrant, endorse, guarantee, or assume responsibility for the accuracy or reliability of any information offered by third-party websites mentioned in any documents or source code. I will not be a party to or in any way be responsible for monitoring any transaction between you and third-party providers of products or services.

BY USING THESE WEBSITES (<https://tech2talk.github.io/vms-and-containers/> or <https://github.com/tech2talk/vms-and-containers>) OR THE INFORMATION CONTAINED ON THEM OR WITHIN ANY DOCUMENTS STORED IN THESE SITES INCLUDING BUT NOT LIMITED TO SOURCE CODE, YOU AGREE TO THE “DISCLAIMER” (INCLUDING “EXTERNAL LINKS DISCLAIMER”) SPECIFIED IN ALL OF THE PREVIOUS PAGES AS WELL AS THE CURRENT PAGE OF THIS DOCUMENT.

This disclaimer is also available at the following link.

<https://tech2talk.github.io/vms-and-containers/disclaimer.pdf>

1. Open a Terminal on Ubuntu and then copy or type the following command
`sudo apt-get install docker.io docker-compose`
2. Enter your password that you created earlier during Ubuntu install and then press enter.
3. Enter “Y” when shell prompts to do so and then press enter.

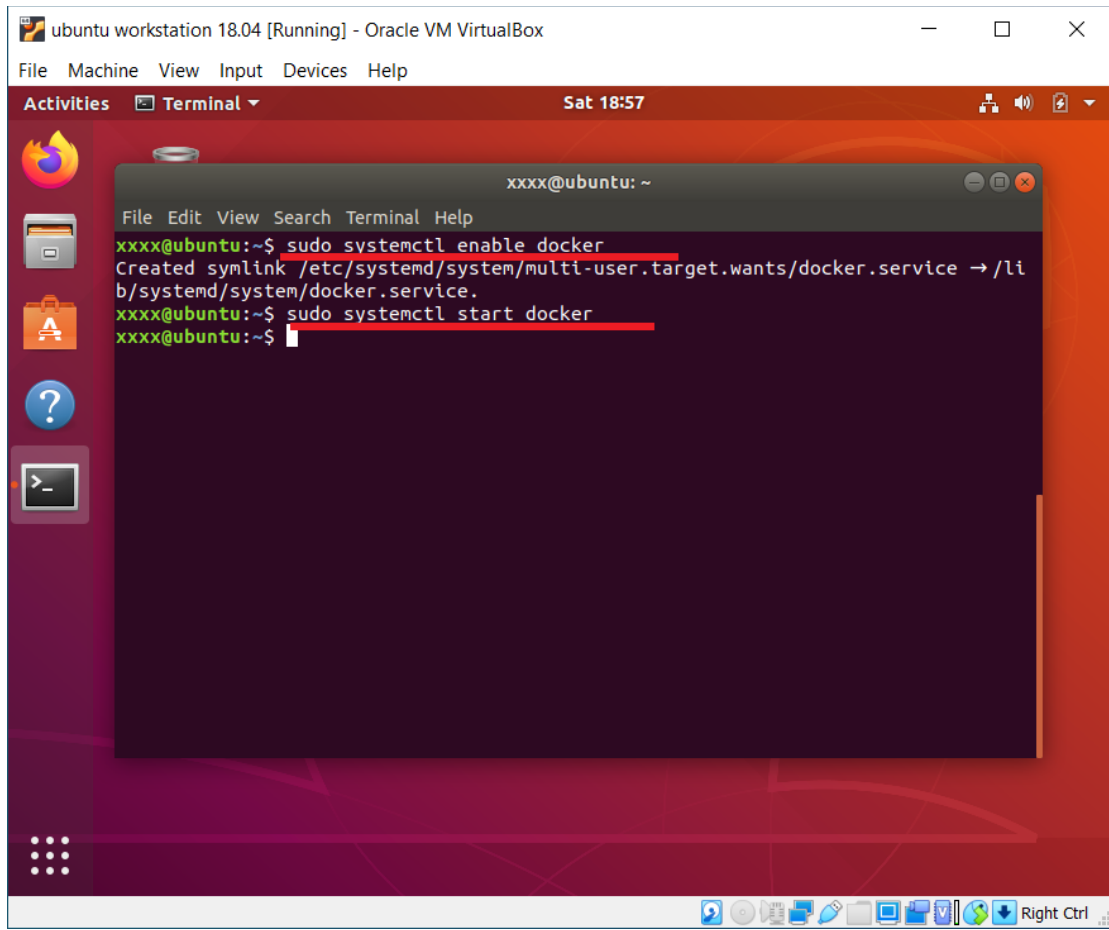


The screenshot shows a terminal window titled 'xxxxx@ubuntu: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal output displays a list of suggested packages, followed by the packages to be installed. It then shows the disk space requirements and asks for confirmation to continue. The user has entered 'Y'.

```
python-urllib3 python-websocket python-yaml python2.7 python2.7-minimal runc
ubuntu-fan
Suggested packages:
aufs-tools btrfs-progs debootstrap docker-doc rinse zfs-fuse | zfsutils
git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
gitweb git-cvs git-mediawiki git-svn python-doc python-tk
python-cryptography-doc python-cryptography-vectors python-enum34-doc
python-funcsigs-doc python-mock-doc python-openssl-doc python-openssl-dbg
python-setuptools python-socks python-ntlm python2.7-doc binfmt-support
The following NEW packages will be installed:
bridge-utils cgroupfs-mount containerd docker-compose docker.io git git-man
golang-docker-credential-helpers liberror-perl libpython-stdlib pigz python
python-asn1crypto python-backports.ssl-match-hostname python-cached-property
python-certifi python-ccfi-backend python-chardet python-cryptography
python-docker python-dockerpty python-dockerpycreds python-docopt
python-enum34 python-funcsigs python-functools32 python-idna
python-ipaddress python-jsonschema python-minimal python-mock python-openssl
python-pbr python-pkg-resources python-requests python-six python-texttable
python-urllib3 python-websocket python-yaml python2.7 python2.7-minimal runc
ubuntu-fan
0 upgraded, 44 newly installed, 0 to remove and 1 not upgraded.
Need to get 70.7 MB/72.3 MB of archives.
After this operation, 368 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

4. Once the install is done, copy or type the following commands into the same terminal one after the other. Let the first command finish before you type the next one.

```
sudo systemctl enable docker
sudo systemctl start docker
```

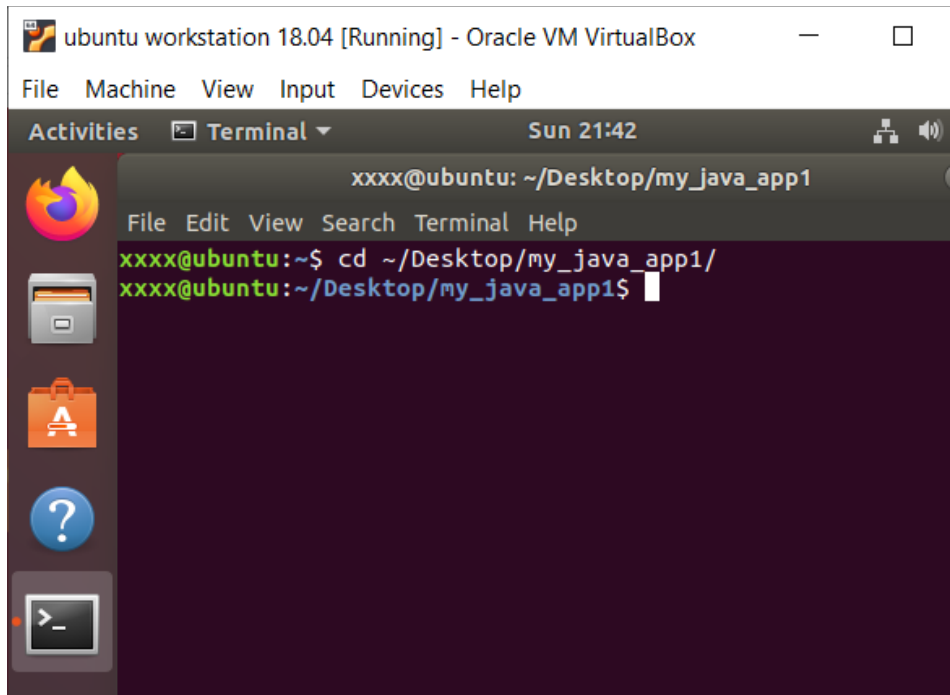


5. Let's create our first Docker container for a Java app.
Create a new folder on desktop of Ubuntu VM and name it **my_java_app1**



6. Switch to the Terminal window on Ubuntu and navigate to the newly created folder by typing or copying the following command into terminal.

```
cd ~/Desktop/my_java_app1/
```



7. We need to download two files for creating our container app.
Copy or type the following command on the terminal and hit enter.

```
wget https://raw.githubusercontent.com/tech2talk/vms-and-containers/main/code/Dockerfile
```

It will download Dockerfile into `my_java_app1` folder.

8. Copy or type the following command on the terminal and hit enter. This will download the second file needed of our app.

```
wget https://github.com/tech2talk/vms-and-containers/raw/main/code/helloworld.jar
```

```

xxxx@ubuntu:~/Desktop/my_java_app1
File Edit View Search Terminal Help
xxxx@ubuntu:~/Desktop/my_java_app1$ wget https://raw.githubusercontent.com/tech2talk/vms-and-containers/main/code/Dockerfile
--2021-04-04 21:46:17-- https://raw.githubusercontent.com/tech2talk/vms-and-containers/main/code/Dockerfile
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 127 [text/plain]
Saving to: 'Dockerfile'

Dockerfile                               100%[=====] 127
2021-04-04 21:46:17 (2.32 MB/s) - 'Dockerfile' saved [127/127]

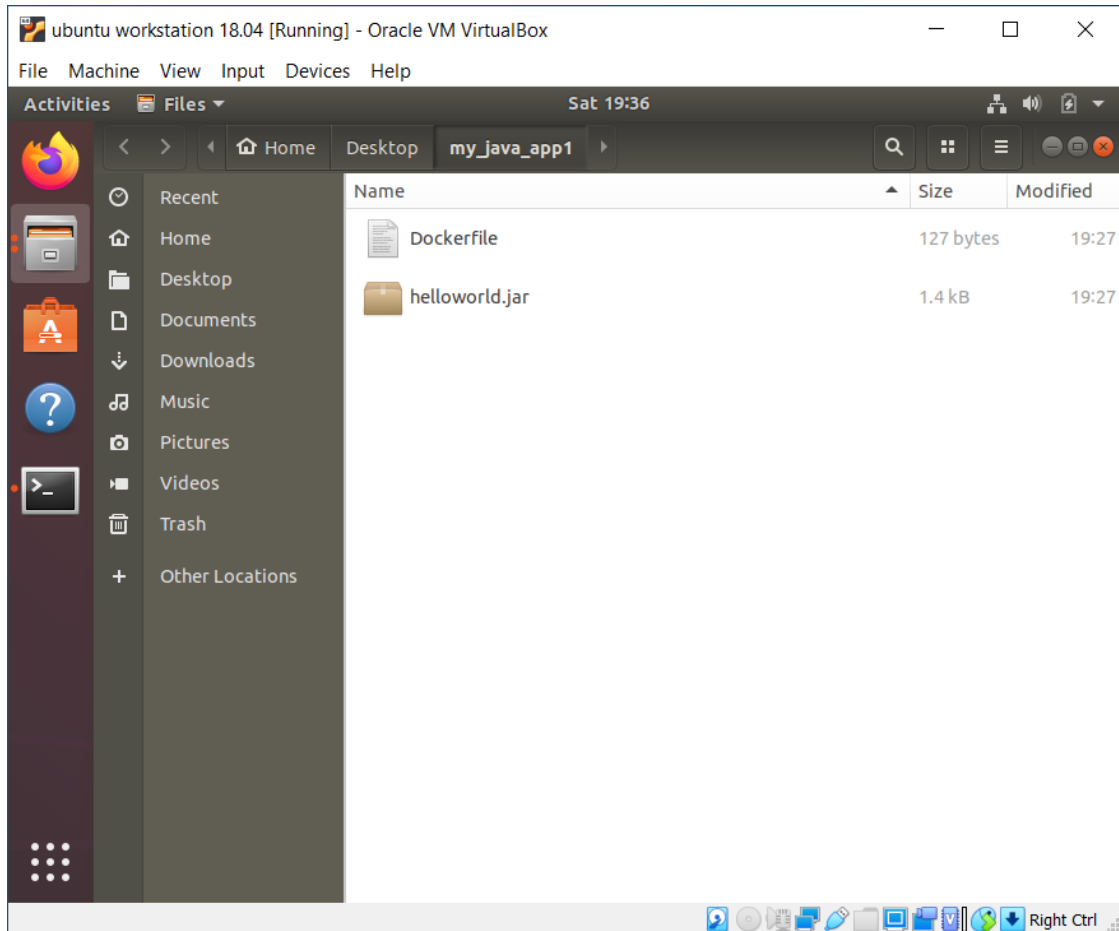
xxxx@ubuntu:~/Desktop/my_java_app1$ wget https://github.com/tech2talk/vms-and-containers/raw/main/code/helloworld.jar
--2021-04-04 21:50:53-- https://github.com/tech2talk/vms-and-containers/raw/main/code/helloworld.jar
Resolving github.com (github.com)... 140.82.114.4
Connecting to github.com (github.com)|140.82.114.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/tech2talk/vms-and-containers/main/code/helloworld.jar [following]
--2021-04-04 21:50:53-- https://raw.githubusercontent.com/tech2talk/vms-and-containers/main/code/helloworld.jar
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.108.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1356 (1.3K) [application/octet-stream]
Saving to: 'helloworld.jar'

helloworld.jar                               100%[=====] 1.32K
2021-04-04 21:50:53 (336 KB/s) - 'helloworld.jar' saved [1356/1356]

xxxx@ubuntu:~/Desktop/my_java_app1$

```

9. After this action, you should see the two files within `my_java_app1` folder.



10. Open a terminal on your Ubuntu and then
Navigate to the following directory structure by typing or copying pasting this command

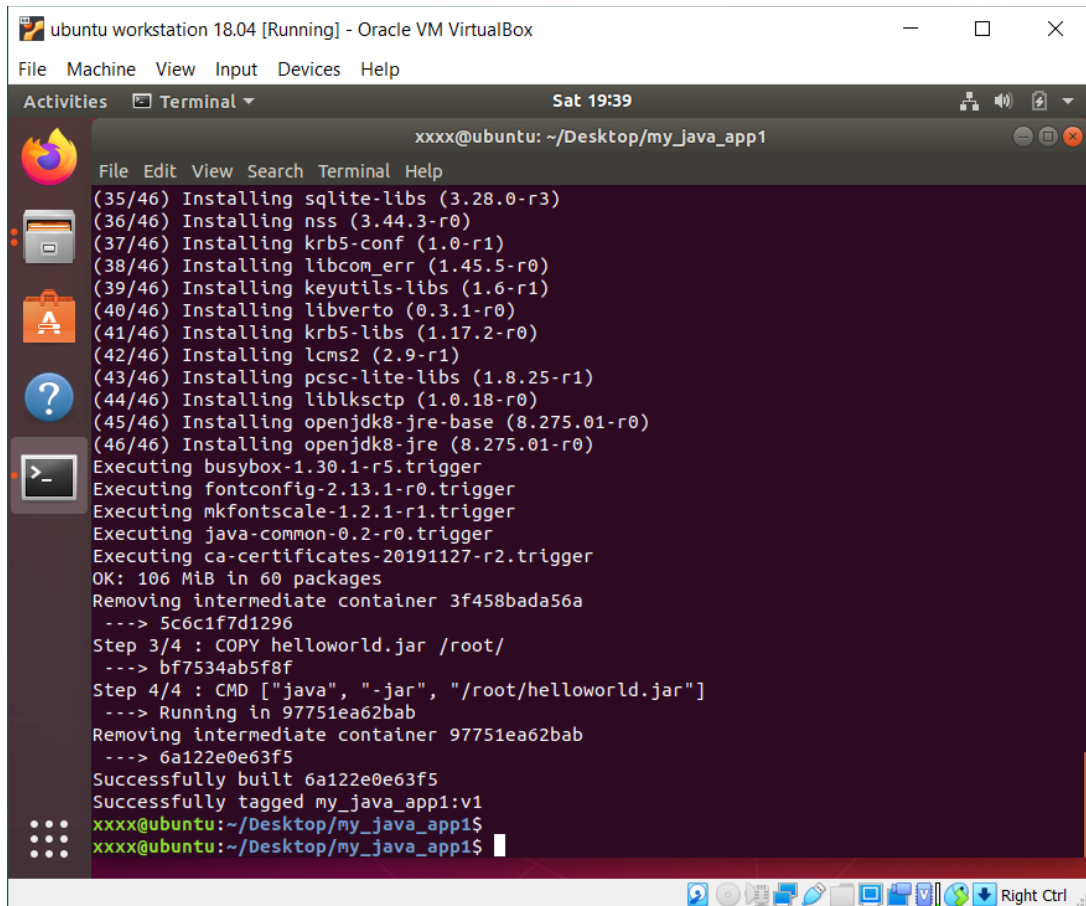
```
cd ~/Desktop/my_java_app1
```

11. Copy or type the following command to build Docker image for your first Java app.

Note: the command has a "." dot at the end so type a "." as well.

```
sudo docker build -t my_java_app1:v1 .
```

12. If everything goes ok then you should see something like this.



```
ubuntu workstation 18.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Sat 19:39
xxxx@ubuntu: ~/Desktop/my_java_app1
File Edit View Search Terminal Help
(35/46) Installing sqlite-libs (3.28.0-r3)
(36/46) Installing nss (3.44.3-r0)
(37/46) Installing krb5-conf (1.0-r1)
(38/46) Installing libcom_err (1.45.5-r0)
(39/46) Installing keyutils-libs (1.6-r1)
(40/46) Installing libverto (0.3.1-r0)
(41/46) Installing krb5-libs (1.17.2-r0)
(42/46) Installing lcms2 (2.9-r1)
(43/46) Installing psc-lite-libs (1.8.25-r1)
(44/46) Installing libksctp (1.0.18-r0)
(45/46) Installing openjdk8-jre-base (8.275.01-r0)
(46/46) Installing openjdk8-jre (8.275.01-r0)
Executing busybox-1.30.1-r5.trigger
Executing fontconfig-2.13.1-r0.trigger
Executing mkfontscale-1.2.1-r1.trigger
Executing java-common-0.2-r0.trigger
Executing ca-certificates-20191127-r2.trigger
OK: 106 MiB in 60 packages
Removing intermediate container 3f458bada56a
--> 5c6c1f7d1296
Step 3/4 : COPY helloworld.jar /root/
--> bf7534ab5f8f
Step 4/4 : CMD ["java", "-jar", "/root/helloworld.jar"]
--> Running in 97751ea62bab
Removing intermediate container 97751ea62bab
--> 6a122e0e63f5
Successfully built 6a122e0e63f5
Successfully tagged my_java_app1:v1
xxxx@ubuntu:~/Desktop/my_java_app1$
xxxx@ubuntu:~/Desktop/my_java_app1$
```

13. Let's run a container from this image by copying or typing the following command.

```
sudo docker run my_java_app1:v1
```

If all goes well then you should see the message in green from this Docker app.

