

May 21, 2021

How to Build Your First DevOps Lab – Part 3: Setting up our Ubuntu Node

What is Docker and its Advantages?

Docker, at its core, is a virtualization technology. But, if you have worked with other virtualization technology, such as hypervisors, Docker is quite different. Docker uses what are called containers that run on top of the host Operating System (OS). These containers are very lightweight and include only the software needed for the task at hand. Docker containers provide consistency in their deployments, having the same configuration each time given the same Dockerfile is used; A Dockerfile is a set of commands executed when a docker image is created. The ability to provide a consistent environment with a small footprint is extremely valuable in DevOps where the containers may have a short lifespan.

Node Overview and Configuration

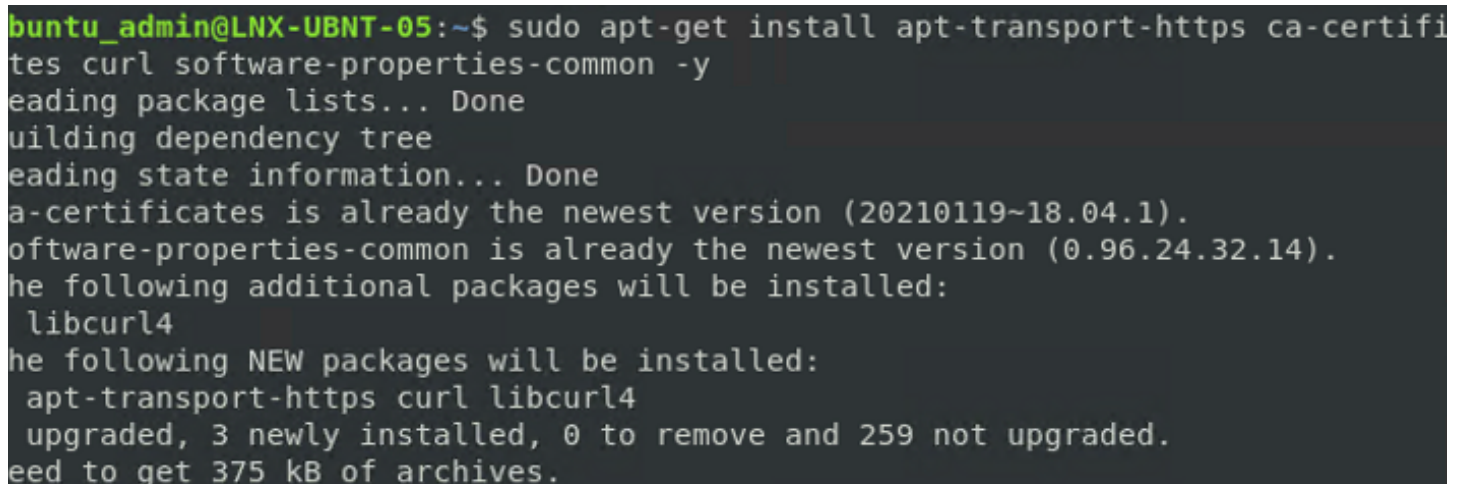
The Ubuntu node in this configuration will be used as a host for Docker containers, as mentioned previously. Running Docker containers on a separate VM will alleviate some compute requirements on the Ubuntu Master VM. While Docker containers do not require a lot of compute, there is already a lot of components being installed on the Ubuntu Master VM.

We will go through the process of installing Docker and configuring SSH to allow connections from the Ubuntu master VM with Ansible. After configuring this node, you likely will not have to do much more on here manually as Ansible can take care of any further configurations remotely.

To configure the Ubuntu node with Docker, see below:

1. Login to Ubuntu Node VM with the configured user
2. Install Docker on Node
 - a. Allow apt to use a repository over HTTPS

```
sudo apt install apt-transport-https ca-certificates curl  
software-properties-common
```



```
buntu_admin@LNx-UBNT-05:~$ sudo apt-get install apt-transport-https ca-certifi  
tes curl software-properties-common -y  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
ca-certificates is already the newest version (20210119~18.04.1).  
software-properties-common is already the newest version (0.96.24.32.14).  
The following additional packages will be installed:  
  libcurl4  
The following NEW packages will be installed:  
  apt-transport-https curl libcurl4  
3 upgraded, 3 newly installed, 0 to remove and 259 not upgraded.  
Need to get 375 kB of archives.
```

b. Add the Docker GPG Key to the list of keys used by apt to authenticate packages

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -
```

```
buntu_admin@LNx-UBNT-05:~$ curl -fsSL https://download.docker.com/linux/ubuntu
pg | sudo apt-key add -
K
```

c. Add Docker repository

```
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu bionic stable"
```

```
buntu_admin@LNx-UBNT-05:~$ sudo add-apt-repository "deb [arch=amd64] https://d
nload.docker.com/linux/ubuntu bionic stable"
it:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
et:2 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
et:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
et:4 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
et:5 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
it:6 http://ppa.launchpad.net/martinx/xrdp-hwe-18.04/ubuntu bionic InRelease
et:7 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Meta
ta [295 kB]
```

d. Update apt to make use of the new repository

```
sudo apt update
```

```
buntu_admin@LNx-UBNT-05:~$ sudo apt update
it:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
it:2 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
it:3 https://download.docker.com/linux/ubuntu bionic InRelease
it:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
it:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
```

e. Install Docker on the Node

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

```
buntu_admin@LNx-UBNT-05:~$ sudo apt install docker-ce -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-ce-cli docker-ce-rootless-extras git git-man
  liberror-perl pigz
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit
  git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
Recommended packages:
  slirp4netns
The following NEW packages will be installed:
```

f. Add Ubuntu user to Docker group to be able to run the docker command without needing sudo

Note: If needed, update the username with the username in your environment

```
sudo gpasswd -a [username] docker
```

```
buntu_admin@LNX-UBNT-05:~$ sudo gpasswd -a ubuntu_admin docker
sudo] password for ubuntu_admin:
adding user ubuntu admin to group docker
```

g. Check that the docker service is running

```
sudo systemctl status docker
```

```
buntu_admin@LNX-UBNT-05:~$ sudo systemctl status docker
docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Mon 2021-03-01 06:39:00 EST; 5s ago
     Docs: https://docs.docker.com
  Main PID: 28237 (dockerd)
    Tasks: 10
   CGroup: /system.slice/docker.service
           └─28237 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/conta
ar 01 06:38:59 LNX-UBNT-05 dockerd[28237]: time="2021-03-01T06:38:59.744087595
ar 01 06:38:59 LNX-UBNT-05 dockerd[28237]: time="2021-03-01T06:38:59.830286621
ar 01 06:38:59 LNX-UBNT-05 dockerd[28237]: time="2021-03-01T06:38:59.830479936
ar 01 06:38:59 LNX-UBNT-05 dockerd[28237]: time="2021-03-01T06:38:59.830672016
```

3. Install Pip

a. Pip is used to install Python packages that can then be used, imported, when running python scripts or running commands directly from the Python prompt

```
sudo apt-get install python3-pip -y
```

```
buntu_admin@LNX-UBNT-05:~$ sudo apt-get install python3-pip -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  build-essential dh-python dpkg-dev fakeroot g++ g++-7 gcc gcc-7
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libasan4 libatomic1 libc-dev-bin libc6 libc6-dbg libc6-dev libcilkrts5
  libdpkg-perl libexpat1-dev libfakeroot libgcc-7-dev libitm1 liblsan0 libmpx2
  libpython3-dev libpython3.6-dev libquadmath0 libstdc++-7-dev libtsan0
  libubsan0 linux-libc-dev make manpages-dev python-pip-whl python3-dev
  python3-distutils python3-lib2to3 python3-setuptools python3-wheel
  python3.6-dev
```

4. Install docker-py to be able to deploy docker machines from Ansible

```
sudo pip3 install docker-py
```



```
buntu_admin@LNX-UBNT-05:~$ sudo pip3 install docker-py
Collecting docker-py
  Downloading https://files.pythonhosted.org/packages/23/c7/1fd6d4d620809fe2f3269d719e2dd0086c939b67021303a9ec40f5a05b/docker_py-1.10.6-py2.py3-none-any.whl (0kB)
    100% |#####| 51kB 2.0MB/s
Requirement already satisfied: requests!=2.11.0,>=2.5.2 in /usr/lib/python3/dist-packages (from docker-py)
Collecting docker-pycreds>=0.2.1 (from docker-py)
  Downloading https://files.pythonhosted.org/packages/f5/e8/f6bd1eee09314e7e6de9cbe2c5e22314ccdb38db16c9fc72d2fa80d054/docker_pycreds-0.4.0-py2.py3-none-any.whl (10kB)
```

5. Install Open SSH Server to allow SSH connections from the Ubuntu Master VM

```
sudo apt-get install openssh-server -y
```

```
buntu_admin@LNX-UBNT-05:~$ sudo apt-get install openssh-server -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere rssh ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 256 not upgraded.
Need to get 637 kB of archives.
```

6. Update Authorized Keys file on Node with Public Keys from Master (These keys should have been copied to an external editor in the part 2)

- Edit the `authorized_keys` file on the Node (This file will get created when running the below command)
- Insert copied contents into a new line and save the file

```
sudo vi ~/.ssh/authorized_keys
```

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCoFsRpMQwn0qhszHpiL6vfm3enkbRfwMrRKBwWI31
CAD4Ryj+S949df1bwN+BA5d8YmGl9Q+kzzzt/dNHX21LrseeX4+YUe5aMHSwtshSQuq0TOZW7Sope

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCoFsRpMQwn0qhszHpiL6vfm3enkbRfwMrRKBwWI31
CAD4Ryj+S949df1bwN+BA5d8YmGl9Q+kzzzt/dNHX21LrseeX4+YUe5aMHSwtshSQuq0TOZW7Sope
```

7. On the Ubuntu Master VM, verify Ansible connectivity to the Ubuntu Node VM

```
ansible all -i /etc/ansible/inventory -m ping
```

```
buntu_admin@LNX-UBNT-04:~/Downloads$ ansible all -m ping -i /etc/ansible/inven
ry
he authenticity of host '172.16.24.138 (172.16.24.138)' can't be established.
CDSA key fingerprint is SHA256:j+ZfdXcb4TSCLR/JiFLS95urMkjMgLrRUTc4LSjwCV0.
re you sure you want to continue connecting (yes/no)? yes
172.16.24.138 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
```

Halfway There

If you have made it this far, you are halfway through the DevOps lab configuration. Congrats! In the following parts, we'll take a look at Jira and Splunk.

Jira is a widely used tool to track issues with different projects for an organization. It helps with managing bugs/issues that may occur in the CI/CD pipeline.

Splunk, on the other hand, is a logging tool used by many organizations for gathering information from remote systems. Splunk doesn't tie together with any of the other components that we configure in this lab build, but it is a great tool to learn nonetheless; however, you could integrate it with other products if you wanted to.

Be on the lookout for Part 4 in this series!

DevOps Lab Build Series Index

[Part 1: Introduction](#)

[Part 2: Configuring the Ubuntu Master](#)

[Part 3: Setting up our Ubuntu Node](#)

[Part 4: Configuring Splunk](#)

[Part 5: Jira Cloud Integration](#)