Linux

All things Linux OS

- Ubuntu Netplan How-to
- Automate Install of Docker on Linux Using Ansible
- Ubuntu Root LVM Volume Resize

Ubuntu Netplan How-to

Netplan DHCP Example:

```
network:
    version: 2
    renderer: networkd
    ethernets:
        enp3s0:
            dhcp4: true
```

Netplan Static Example:

```
network:
    version: 2
    renderer: networkd
    ethernets:
        enp3s0:
            addresses:
                - 10.10.10.2/24
            nameservers:
                search: [mydomain, otherdomain]
                addresses: [10.10.10.1, 1.1.1.1]
            routes:
                - to: default
                  via: 10.10.10.1
```

Automate Install of Docker on Linux Using Ansible

Step 1: Get the Docker Host Ready

In this first step the host has to be prepared to work with ansible. In this example, Ubuntu 22.04 is installed on the host with all packages up to date. It is assumed that Ansible is already installed on a controller computer. This document does not walk through the how-to of installing and the initial configuration of Ansible.

1. Add a user to the host that will be used by Ansible
2. Set password for newly created user
3. Ensure the user is a member of the sudoers group
4. Modify the sudoers config to allow the user to bypass sudo password
5. Make sure host entry is included in the /etc/hosts file so that Ansible can reach it by the hostname
6. Create SSH key on Ansible controller for connectivity to host
7. Copy created ssh key to the host
8. Test the key

```
gtaylor@dockerhost:~$ sudo useradd -m ansible
[sudo] password for gtaylor:
gtaylor@dockerhost:~$ sudo passwd ansible
New password:
Retype new password:
passwd: password updated successfully
```

```
gtaylor@dockerhost:~$ sudo usermod -aG sudo ansible
gtaylor@dockerhost:~$ groups ansible
ansible : ansible sudo
gtaylor@dockerhost:~$
```

```
gtaylor@dockerhost:~$ sudo visudo
```

Add the following line to the bottom of the sudoers file to allow the ansible user to sudo without prompting for a password. Warning this can be dangerous be sure to protect the ansible user and the keys we will be using to login within the next section.

ansible ALL=(ALL) NOPASSWD:ALL

On the Ansible controller check the hosts file (/etc/hosts) to ensure that the dockerhost server is listed with IP address. I am using WSL2 for Windows, so I actually added the host entry in the Windows host file (C:\Windows\System32\drivers\etc\host). Note the comment at the top of the /etc/hosts file in WSL, it is generated automatically from the Windows side.

```
gtaylor@lpt2:~$ cat /etc/hosts
# This file was automatically generated by WSL. To stop automatic generation of this file, add the following
entry to /etc/wsl.conf:
# [network]
# generateHosts = false
127.0.0.1       localhost
127.0.1.1       lpt2.localhost     lpt2
10.2.3.100      dockerhost

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Next, it's time to generate the ssh key for the Ansible controller to connect without a password to the dockerhost server. DO NOT ENTER A PASSPHRASE when prompter, just hit enter to bypass the passphrase.

```
gtaylor@dockerhost:~$ ssh-keygen -t ed25519 -C Ansible
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/gtaylor/.ssh/id_ed25519): /home/gtaylor/.ssh/ansible_ed25519
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
Your identification has been saved in /home/gtaylor/.ssh/ansible_ed25519
Your public key has been saved in /home/gtaylor/.ssh/ansible_ed25519.pub
The key fingerprint is:
SHA256:p+QG57C3PRgVFsiu/AMdqtUkmhnDFIidEiOkWLUqsL0 Ansible
The key's randomart image is:
+--[ED25519 256]--+
|==.+o. . ...     |
|*o+ ..  o o      |
|+. o.  . . .     |
|.o .+ . + .      |
|o o  BoBSo.      |
| . .+ *O+o       |
|  E  o.o*o       |
|    .  o+o.      |
|       ....      |
+----[SHA256]-----+
```

```
gtaylor@dockerhost:~$ ssh-copy-id -i /home/gtaylor/.ssh/ansible_ed25519.pub ansible@dockerhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/gtaylor/.ssh/ansible_ed25519.pub"
The authenticity of host 'dockerhost (127.0.1.1)' can't be established.
ED25519 key fingerprint is SHA256:w1J+iIIXLt6STUqVgGnxqa423M/BO7pNDOAc4FJoxvM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@dockerhost's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ansible@dockerhost'"
and check to make sure that only the key(s) you wanted were added.
```

Test the connection. Be sure to include the correct key in the ssh connection command.

```
gtaylor@lpt2:/mnt/c/Users/gtaylor/Documents/ansible/projects$ ssh -i /home/gtaylor/.ssh/ansible_ed25519
ansible@dockerho
st
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-52-generic x86_64)
```

```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Nov 14 02:16:27 PM UTC 2022

  System load:  0.0               Processes:             213
  Usage of /:   29.8% of 9.75GB   Users logged in:       1
  Memory usage: 10%               IPv4 address for ens33: 10.2.3.100
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

0 updates can be applied immediately.



$ exit
Connection to dockerhost closed.
```

Step 2: Prepare Ansible

The next step is to prepare Ansible for the installation.

1. Get the Ansible inventory file ready
2. Get the Ansible playbook file created

Make sure you have your Ansible inventory file created. For more information on creating an inventory check out the Ansible documentation.

Next, create the Ansible playbook for the installation of the Docker prerequisites, Docker, Docker Compose, and Docker SDK.

The playbook I used below was adapted from Christian Lempa's playbook. The only thing I modified is I changed the hosts declaration to point to my dockerhost server and I added a variable for my ssh private key file location.

```
---

- hosts: dockerhost
  become: yes
  vars:
    ansible_ssh_private_key_file: "/home/gtaylor/.ssh/ansible_ed25519"

  tasks:
  # Install Docker
  # --
  #
  - name: install prerequisites
    apt:
      name:
        - apt-transport-https
        - ca-certificates
        - curl
        - gnupg-agent
        - software-properties-common
      update_cache: yes

  - name: add apt-key
    apt_key:
      url: https://download.docker.com/linux/ubuntu/gpg

  - name: add docker repo
    apt_repository:
      repo: deb https://download.docker.com/linux/ubuntu focal stable

  - name: install docker
    apt:
      name:
        - docker-ce
        - docker-ce-cli
        - containerd.io
      update_cache: yes

  - name: add user permissions
    shell: "usermod -aG docker {{ ansible_env.SUDO_USER }}"
```

```
  - name: Reset ssh connection for changes to take effect
    meta: "reset_connection"

  # Installs Docker SDK
  # --
  #
  - name: install python package manager
    apt:
      name: python3-pip

  - name: install python sdk
    become_user: "{{ ansible_env.SUDO_USER }}"
    pip:
      name:
        - docker
        - docker-compose
```

Step 3: Install Docker on Host Using Ansible

Now it's time to put it all together and install Docker using Ansible. To do this we use ansible-playbook to apply the playbook for installing Docker on the host.

Run the playbook

```
gtaylor@lpt2:/mnt/c/Users/gtaylor/Documents/ansible/projects$ ansible-playbook install_docker.yml

PLAY [dockerhost]
*********************************************************************************************************

TASK [Gathering Facts]
*********************************************************************************************************
ok: [dockerhost]

TASK [install prerequisites]
*********************************************************************************************
changed: [dockerhost]
```

```
TASK [add apt-key]
*********************************************************************************************************
changed: [dockerhost]

TASK [add docker repo]
*********************************************************************************************************
changed: [dockerhost]

TASK [install docker]
*********************************************************************************************************
changed: [dockerhost]

TASK [add user permissions]
*********************************************************************************************
changed: [dockerhost]

TASK [Reset ssh connection for changes to take effect]
*************************************************************

TASK [install python package manager]
*************************************************************************************
changed: [dockerhost]

TASK [install python sdk]
*********************************************************************************************
changed: [dockerhost]

PLAY RECAP
*********************************************************************************************************************
dockerhost                 : ok=8    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

SSH to the docker host as the ansible user, again be sure to declare the correct ssh key for ansible user, and check that Docker was installed.

```
$ docker version
Client: Docker Engine - Community
 Version:           20.10.21
 API version:       1.41
 Go version:        go1.18.7
 Git commit:        baeda1f
```

```
 Built:             Tue Oct 25 18:02:21 2022
 OS/Arch:           linux/amd64
 Context:           default
 Experimental:      true

Server: Docker Engine - Community
 Engine:
  Version:          20.10.21
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.18.7
  Git commit:       3056208
  Built:            Tue Oct 25 18:00:04 2022
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.9
  GitCommit:        1c90a442489720eec95342e1789ee8a5e1b9536f
 runc:
  Version:          1.1.4
  GitCommit:        v1.1.4-0-g5fd4c4d
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
$
```

Docker is installed along with the other components. Happy containering!

Ubuntu Root LVM Volume Resize

```
lsblk

sudo lvextend -l +100%FREE /dev/mapper/ubuntu--vg-ubuntu--lv

sudo resize2fs /dev/mapper/ubuntu--vg-ubuntu--lv

lsblk
```