Details Installation step in VM

Cloud Essentials (Lovely Professional University)

Step 1. **Creating a VM (using VMware Workstation or VirtualBox)**

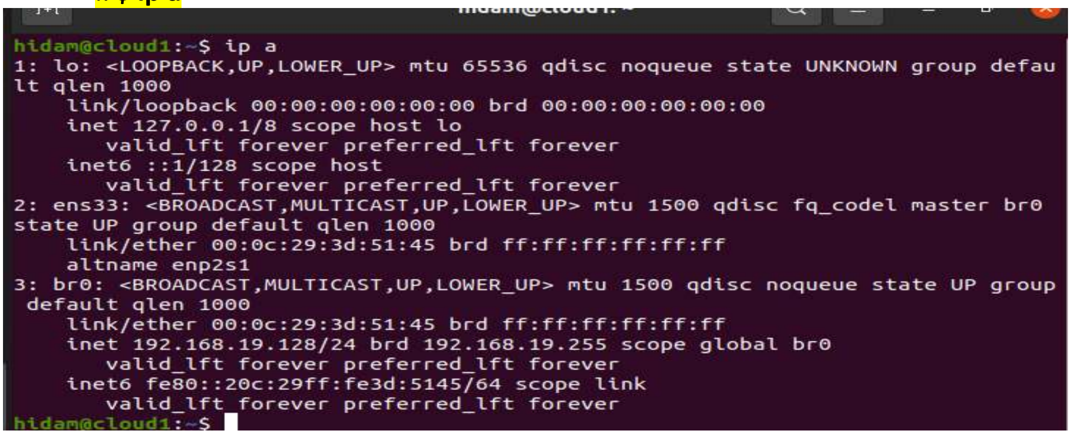
a. Create a VM using either VMware or VirtualBox

- Configuration:
    - 4GB ram (Min 3)
    - 40 GB+ storage
    - NAT Adapter
- **Install Ubuntu OS (Ubuntu 20 or above)**

Step 2. **Check the IP address of the Guest OS:**

a. Open a terminal and run the following Commands:

i. **$ ip a**

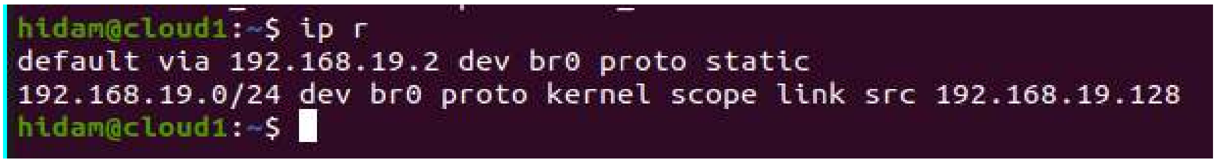


```
hidam@cloud1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master br0
state UP group default qlen 1000
    link/ether 00:0c:29:3d:51:45 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
3: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
 default qlen 1000
    link/ether 00:0c:29:3d:51:45 brd ff:ff:ff:ff:ff:ff
    inet 192.168.19.128/24 brd 192.168.19.255 scope global br0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe3d:5145/64 scope link
       valid_lft forever preferred_lft forever
hidam@cloud1:~$
```

From the given output:

check for interface: **ens33** (in the case of about screenshot)

ii. $ip r



```
hidam@cloud1:~$ ip r
default via 192.168.19.2 dev br0 proto static
192.168.19.0/24 dev br0 proto kernel scope link src 192.168.19.128
hidam@cloud1:~$
```

From the above screenshot:

- Your gateway is the IP address associated with default via (192.168.19.2)
- Your IP address is the IP address associated with src (192.168.19.128)

Step 3. Set the static IP using netplan configuration file

sudo nano /etc/netplan/01-network-manager-all.yaml

(Note: check the /etc/netplan file available in your system using $ls /etc/netplan)

In the Editor add the following configuration: (Check the indention) and save the file (Ctrl+x)

```
network:
    ethernets:
         enp0s3:
            dhcp4: false
            addresses: [192.168.19.128/24]    Note: change ip address
            gateway4: 192.168.19.2    Note: change ip address
            nameservers:
                    addresses: [8.8.8.8,8.8.4.4]
```

```
#version: 2
```

- Apply the changes: $ sudo netplan apply
- Restart the network manager: $ sudo systemctl restart NetworkManager

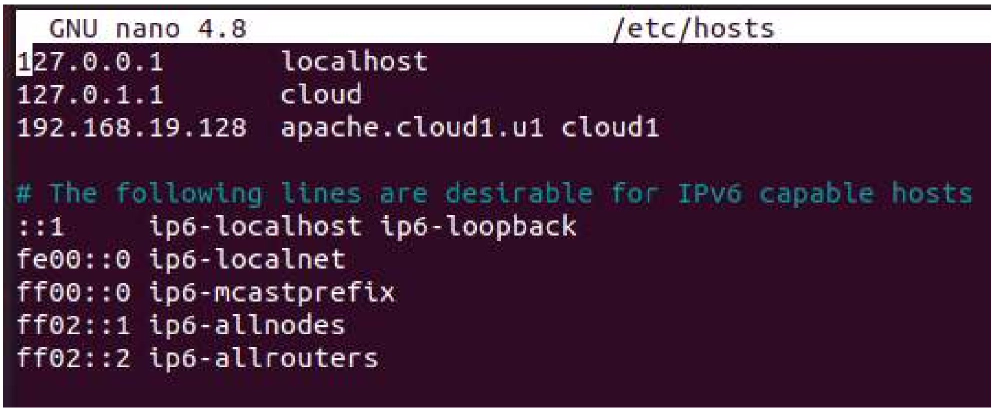
Step 4. Check the hostname : $hostname --fqdn

(Check the hostname shown)

Step 5. Add cloudstack hostname in the host file: $ sudo nano /etc/hosts

Enter the hostname : <static ip> <domain_name> <hostname>

Save the file



```
  GNU nano 4.8                        /etc/hosts
127.0.0.1       localhost
127.0.1.1       cloud
192.168.19.128  apache.cloud1.u1 cloud1

# 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
```

Step 6. Set the Hostname : $sudo hostnamectl set-hostname <hostname>

Step 7. Install bridge-utils: $sudo apt install bridge-utils

Step 8. Add bridge

a. Open A new terminal (Alt+Ctrl+T)

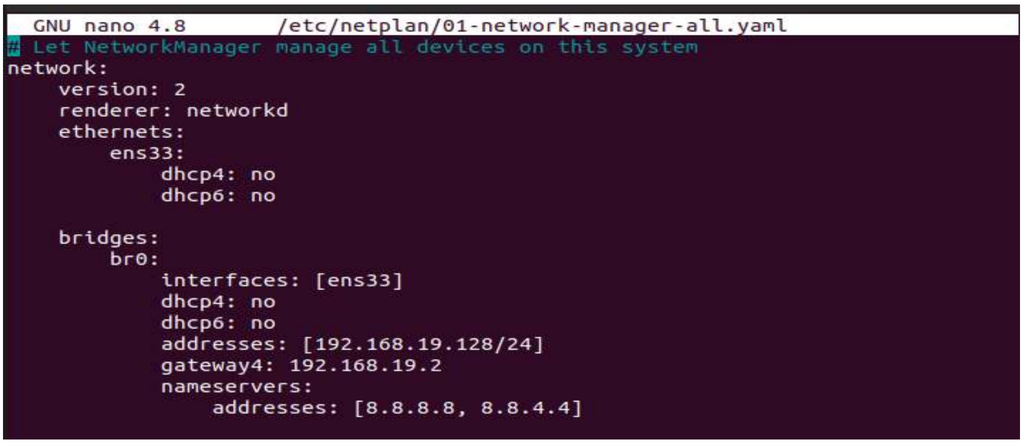
b. Run the commands:

- $brctl addbr br0
- $sudo brctl addif br0 ens33 (ens33 is the adapter-put ths adapter available in your system)

Step 9. In the first terminal open the network-manager file and add the bridge configuration:

a. $sudo nano /etc/netplan/01-network-manager-all.yaml

(Check the indention)



```
  GNU nano 4.8          /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
    version: 2
    renderer: networkd
    ethernets:
        ens33:
            dhcp4: no
            dhcp6: no

    bridges:
        br0:
            interfaces: [ens33]
            dhcp4: no
            dhcp6: no
            addresses: [192.168.19.128/24]
            gateway4: 192.168.19.2
            nameservers:
                addresses: [8.8.8.8, 8.8.4.4]
```

Step 10. $sudo netplan --debbug apply

Step 11. $sudo systemctl restart NetworkManager

Step 12. Install NTP (network Time Protocol)

a. $sudo apt install ntp

Step 13. Install chrony

a. $sudo apt install chrony

Step 14. Install JDK

a. $sudo apt install openjdk-11-jdk

Step 15. Adding Deb package repository:

a. $sudo nano /etc/apt/sources.list.d/cloudstack.list

b. add the following path in the file:

- deb https://download.cloudstack.org/ubuntu focal 4.18

(focal is for Ubuntu 20)

c. Add the public key to the trusted keys.

```
$ wget -O - https://download.cloudstack.org/release.asc | sudo tee /etc/apt/trusted.gpg.d/cloudstack.asc
$ sudo apt update
```

Step 16. Install cloudstack management

a. $sudo apt install cloudstack-management

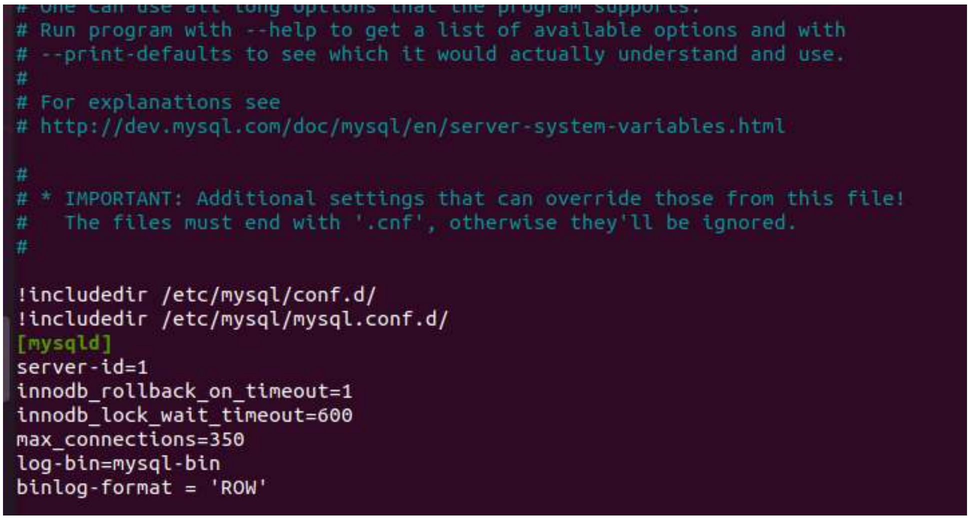
Step 17. Install mysql Server

a. $sudo apt install mysql-server

b. Open my.cnf file and add the configuration

- $sudo nano /etc/mysql/my.cnf
- add the configurations:

```
[mysqld]
server-id=1
innodb_rollback_on_timeout=1
innodb_lock_wait_timeout=600
max_connections=350
log-bin=mysql-bin
binlog-format = 'ROW'
```



```
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

#
# * IMPORTANT: Additional settings that can override those from this file!
#   The files must end with '.cnf', otherwise they'll be ignored.
#

!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/
[mysqld]
server-id=1
innodb_rollback_on_timeout=1
innodb_lock_wait_timeout=600
max_connections=350
log-bin=mysql-bin
binlog-format = 'ROW'
```

Step 18. $sudo systemctl restart mysql

Step 19. Mysql secure installation:

a. $sudo mysql_secure_installation

- Click Yes
- Put 0
- Yes
- yes
- yes
- Yes

Step 20. Run Mysql

a. $sudo mysql

b. in the Mysql interface run the following query:

```
-- Create the cloud and cloud_usage databases
CREATE DATABASE `cloud`;
CREATE DATABASE `cloud_usage`;

-- Create the cloud user
CREATE USER cloud@`localhost` identified by '<password>';
CREATE USER cloud@`%` identified by '<password>';

-- Grant all privileges to the cloud user on the
databases
GRANT ALL ON cloud.* to cloud@`localhost`;
GRANT ALL ON cloud.* to cloud@`%`;

GRANT ALL ON cloud_usage.* to cloud@`localhost`;
GRANT ALL ON cloud_usage.* to cloud@`%`;

-- Grant process list privilege for all other databases
GRANT process ON *.* TO cloud@`localhost`;
GRANT process ON *.* TO cloud@`%`;
```

- exit to the terminal (exit;)

Step 21. Deploy databases for cloudstack:

```
$sudo cloudstack-setup-databases cloud:<root-password>@localhost --deploy-as=root
```

Step 22. run cloudstack management

a. $sudo cloudstack-setup-management

Step 23. allow port for mysql

a. $sudo ufw allow mysql

Step 24. Prepare NFS:

```
$sudo mkdir -p /export/primary

$sudo mkdir -p /export/secondary
```

$sudo nano /etc/exports (insert the commands in the file)

```
/export *(rw,async,no_root_squash,no_subtree_check)
```

Step 25. Install NFS -server:

a. $sudo apt install nfs-kernel-server

```
Step 26.        service nfs-kernel-server restart
                $sudo mkdir -p /mnt/primary
                $sudo mkdir -p /mnt/secondary
```

```
step 27.
echo "192.168.122.10:/export/primary /mnt/primary nfs rsize=8192,wsize=8192,timeo=14,intr,vers=3,noauto 0 2" >> /etc/fstab

echo "192.168.122.10:/export/secondary /mnt/secondary nfs rsize=8192,wsize=8192,timeo=14,intr,vers=3,noauto 0 2" >> /etc/fstab
```

```
                $sudo mount -p /mnt/primary
                $sudo mount -p /mnt/secondary
```

Step 27. Open a browser enter the url to open CLoudstack:

```
http://192.168.19.128:8080/client/               (put your ip address)
```

Note: Default credential is:
Username: **admin**
Password: **password**