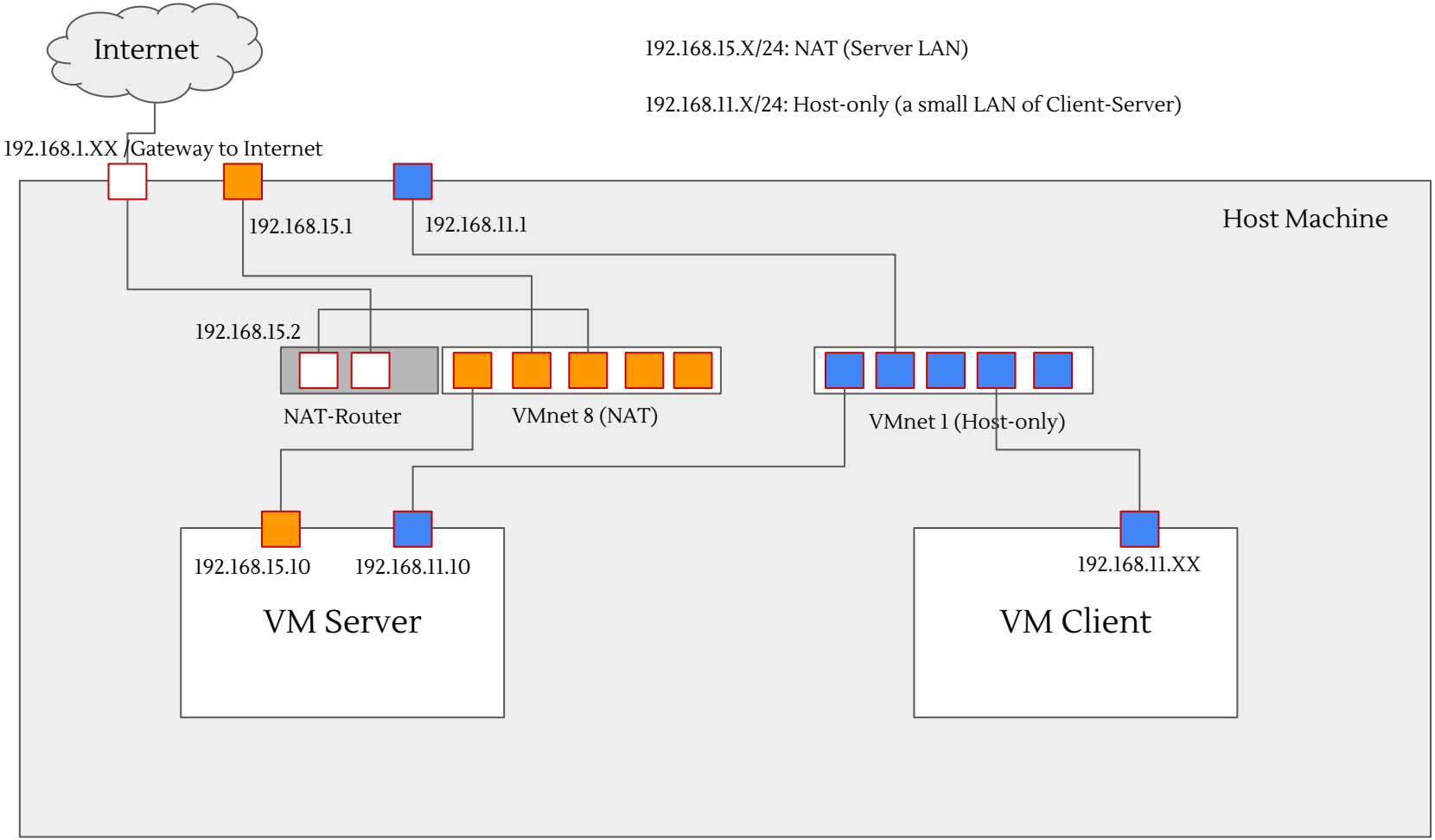

Internet
192.168.15.X/24: NAT (Server LAN)
192.168.11.X/24: Host-only (a small LAN of Client-Server)
192.168.1.XX /Gateway to Internet
192.168.15.1
192.168.11.1
Host Machine
192.168.15.2
NAT-Router
VMnet 8 (NAT)
VMnet 1 (Host-only)
192.168.15.10
192.168.11.10
VM Server
192.168.11.XX
VM Client

1. Environmental setup

Objectives:

- Set up environment in VMWare workstation/ Virtual Box with discussed network IP addresses (2 LANs: Server LAN/ NAT and Client-Server LAN/ Host-only).

✓ In VMWare Workstation: follow instructions in **PDF file: Chapter 1.**

- Install Ubuntu Server on a VM
- Set Static IP address to your Ubuntu Server and make sure it can connect to the Internet just by setting IP address and nameserver addresses.

✓ Use ***ip a*** to show network interface and IP addresses

✓ To set ip address: edit configuration file in ***/etc/netplan/00-installer-config.ymal***

- Set Static IP address to your Ubuntu Server and make sure it can connect to the Internet just by setting IP address and nameserver addresses.

✓ Use *ip a* to show network interface and IP addresses

✓ Switch to root user using: *sudo su*

✓ To set ip address: edit configuration file in */etc/netplan/00-installer-config.ymal*

✓ Run *sudo netplan apply* to apply configuration file.

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens33:
      dhcp4: false
      addresses:
       - 192.168.15.10/24
      gateway4: 192.168.15.2
      nameservers:
         addresses: [8.8.8.8]
```

- *Ping google.com to test connection*

2. Configuring DHCP

1. Install and configure **`isc-dhcp-server`** on the server:

✓ install:

```
sudo apt install isc-dhcp-server
```

✓ Edit config file at `/etc/default/isc-dhcp-server` :

```
sudo nano /etc/default/isc-dhcp-server
```

Add name of network interface that we want it to become DHCP server address (Host-only)

for example if **ens34** is the name of that interface.

`INTERFACEv4="`**`ens34`**`"`

✓ Edit the config file at /etc/dhcp/dhcpd.conf:

```
sudo nano /etc/dhcp/dhcpd.conf
```

Configuring DHCP

- ✓ Add the following lines:

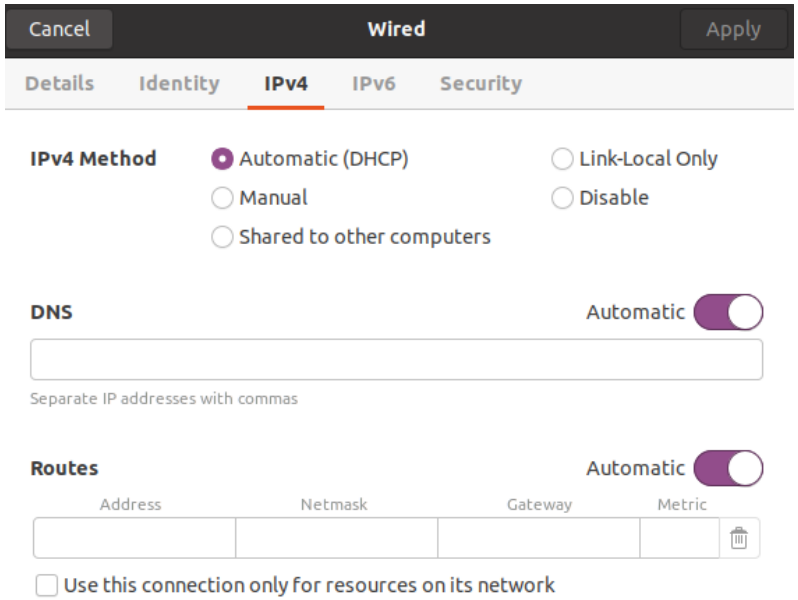
```
subnet 192.168.11.0 netmask 255.255.255.0 {
     range 192.168.11.100 192.168.11.200;
     option domain-name-servers 8.8.8.8;
     option domain-name “google.com”;
     option subnet-mask 255.255.255.0;
     option routers 192.168.12.10;
     default-lease-time 620;
     max-lease-time 7200;
}
```

- ✓ Start the service:

```
sudo systemctl start isc-dhcp-server
```

Configuring DHCP

✓ Go to client and choose DHCP option on its Host-only interface.



✓ Check its IP address to confirm if DHCP works.

3. Configuring Router

1. **Enable packet forwarding:**

✓ Edit config file:

```
sudo nano /etc/sysctl.conf
```

✓ Uncomment :

```
net.ipv4.ip_forward=1
```

✓ Apply change:

```
sysctl -p
```

2. **Configuring Packet Forwarding and IP masquerading:**

```
iptables -A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT
```

`enp0s8` is the name of network interface “Host-only”

`enp0s3` is the name of network interface “NAT”

The above command will forward packets from Host-only interface to NAT interface.

- **Configuring Packet Forwarding and IP masquerading (Continue):**

✓ Now we want to configure packets from NAT interface to be forwarded back to Host-only interface.

Type:

```
iptables -A FORWARD -i enp0s3 -o enp0s8 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

`enp0s8` is the name of network interface “Host-only”

`enp0s3` is the name of network interface “NAT”

✓ Configuring NATing: Now we want to allow all interfaces to use NAT

```
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
```

```
iptables -t nat -A POSTROUTING -o enp0s8 -j MASQUERADE
```

- Saving iptables rules permanently

✓ iptables rule can be reset after reboot. To save rules permanently, install **`iptables-persistent`**

Type:

```
sudo apt install iptables-persistent
```

Switch user to root:

```
sudo su
```

```
iptables-save > /etc/iptables/rules.v4
```

Go to clients and try to ping google.com

4. Installing and Configuring NTP

Objective: time on server and client can be synchronized

1. Install ntp: `sudo apt install ntp`
2. Configure: edit config file `/etc/ntp.conf` and replace the 4 lines of ntp pools

   new pools of Asia timezone server:

```
server  0.asia.pool.ntp.org
server  1.asia.pool.ntp.org
server  2.asia.pool.ntp.org
server  3.asia.pool.ntp.org
```

1. Restart service: `sudo systemctl restart ntp`
2. Check status if it's running : `sudo systemctl status ntp`
3. Configure firewall: `sudo ufw allow 123/udp`

- *Installing and Configuring NTP Client*

GO TO Client

Configuring hostfile

1. Edit hostfile:

```
sudo nano /etc/hosts
```

2. Add to the third line of file:

```
192.168.11.10 servername
```

Configuring NTP

1. `sudo apt install ntp`
2. `sudo /etc/ntp.conf`
3. Replace the 4 lines of ntp pools with:

```
server servername prefer iburst
```

servername is your server hostname. Type command: hostname on your server to check. For example: mine is piseyserver

- *Testing NTP*

✓ Restarting service:

```
sudo systemctl restart ntp
```

✓ Checking synchronized NTP server

```
ntpq -p
```

✓ Testing timezone on both client and server using:

```
timedatectl
```

✓ If it's a right time zone, change it to Asia/Bangkok:

```
timedatectl set-timezone Asia/Bangkok
```

✓ Try changing time on client and restart NTP again. Check if the time returns to its correct synchronized time.

5. Install and Configure DNS Server

- In this practice, we will install DNS using BIND package and configure our server as Primary DNS server, and caching name server as well.

✓ Install bind9 and dnsutils:

Type:

```
sudo apt install bind9 dnsutils
```

- Configure Caching name server:

Open /etc/bind/named.conf.options enable the forwarders section,

and add your preferred DNS servers:

```
forwarders {
    8.8.8.8;
};
```

Testing it by running this 2 times and check the query time:

```
dig -x 8.8.8.8
```

Install and Configure DNS Server

- ✓ Result of query should be faster at 2nd time:

```
;; ANSWER SECTION:
8.8.8.8.in-addr.arpa.   20728   IN      PTR     dns.google.

;; Query time: 36 msec
```

```
;; ANSWER SECTION:
8.8.8.8.in-addr.arpa.   7192    IN      PTR     dns.google.

;; Query time: 0 msec
```