

Thai Blockchain Infrastructure Installation Instructions

Version 1.2

2022-10-20

1. Hardware Requirements

1.1. Minimum

- 1.1.1. CPU x86-64 2 core
- 1.1.2. RAM 4 GB
- 1.1.3. Disk 256 GB SSD
- 1.1.4. 100 Mbps NIC

1.2. Recommended

- 1.2.1. CPU x86-64 4 core
- 1.2.2. RAM 8 GB
- 1.2.3. Disk 512 GB SSD
- 1.2.4. 1 Gbps NIC

New Virtual Machine

1 Select a creation type
2 Select a name and folder
3 Select a compute resource
4 Select storage
5 Select compatibility
6 Select a guest OS
7 Customize hardware
8 Ready to complete

Customize hardware
Configure the virtual machine hardware

Virtual Hardware VM Options

ADD NEW DEVICE

> CPU *	2	
> Memory *	4	GB
> New Hard disk *	250	GB
> New SCSI controller *	LSI Logic Parallel	
> New Network *	vlan121-202.28.118.128/25	<input checked="" type="checkbox"/> Connect...
> New CD/DVD Drive *	Datastore ISO File	<input checked="" type="checkbox"/> Connect...
> Video card *	Specify custom settings	
> Security Devices	Not Configured	
> VNCI device		
> New SATA Controller	New SATA Controller	
> Other	Additional Hardware	

Compatibility: VmHardware.HWVersion.long.vmx-19

CANCEL BACK NEXT

VM Configuration

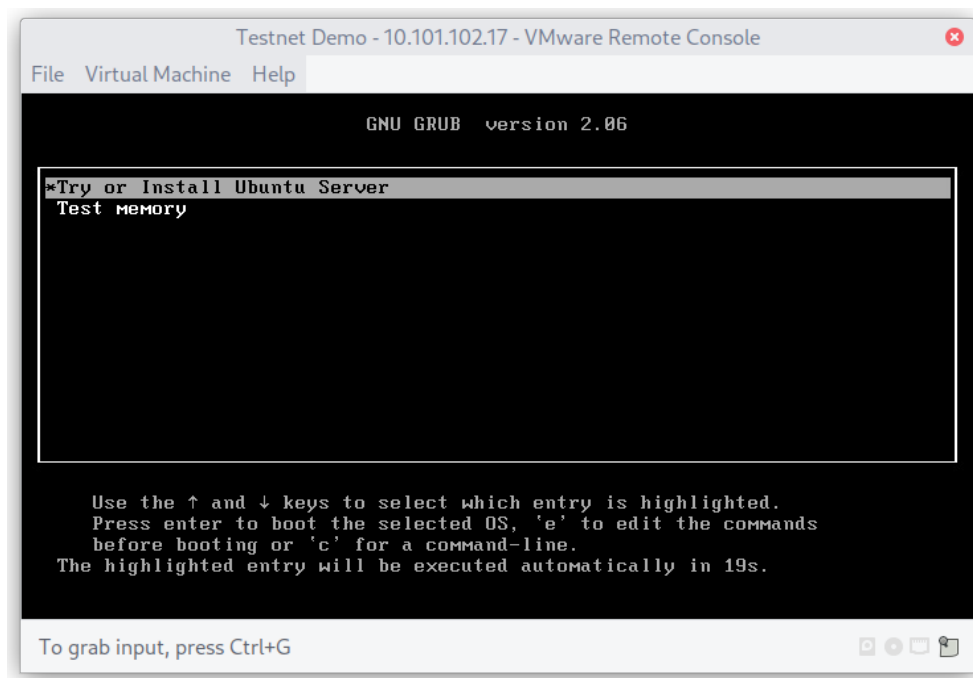
2. เลือก Linux Distribution สำหรับ server ที่สามารถติดตั้ง Docker อปเกรด/แพตช์เร็ว

2.1. Recommend

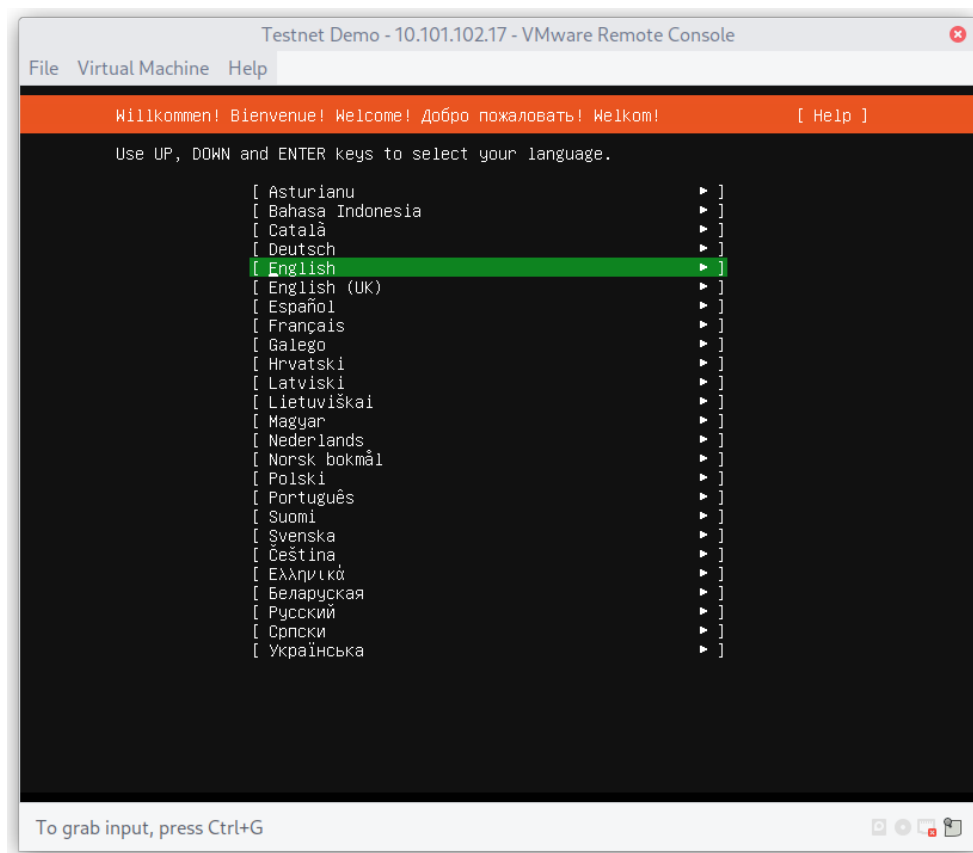
- 2.1.1. Ubuntu Linux 20.04 LTS, 22.04 LTS
<https://mirror.kku.ac.th/ubuntu-releases/>
- 2.1.2. Rocky Linux
<https://mirror.kku.ac.th/rocky-linux/>
- 2.1.3. Alma Linux
<https://mirror.kku.ac.th/almaLinux/>

3. ติดตั้ง Linux Distribution และ basic hardening

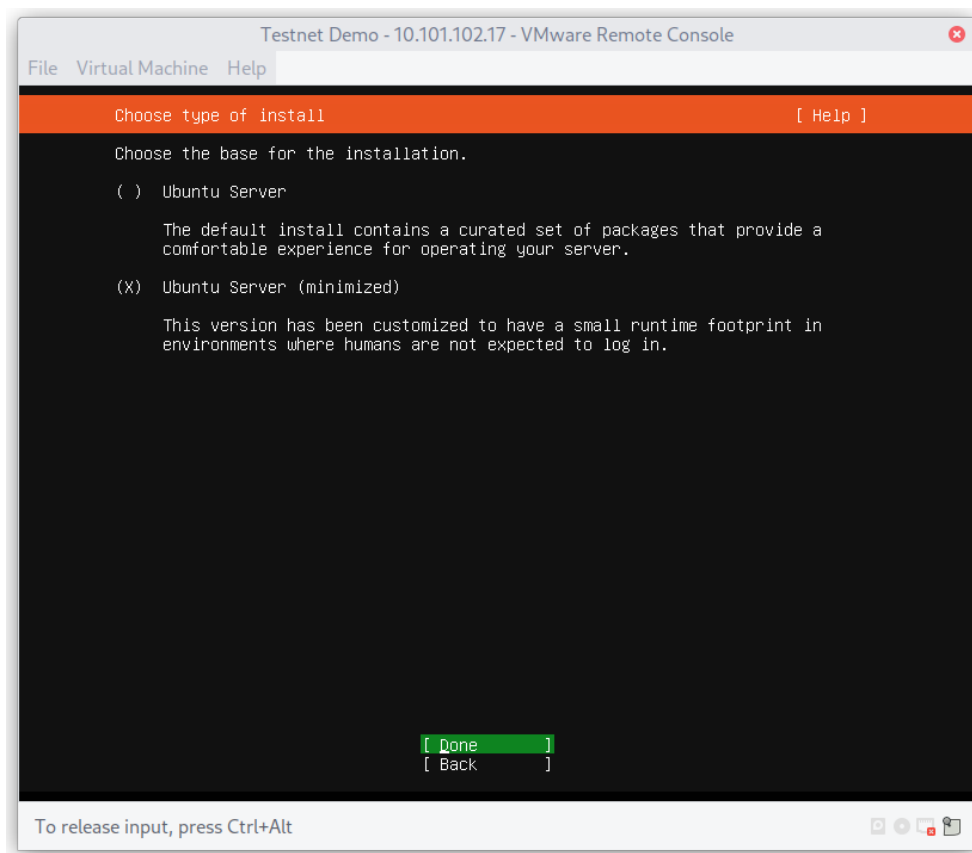
3.1. Text-based, Command Line Interface เท่านั้น



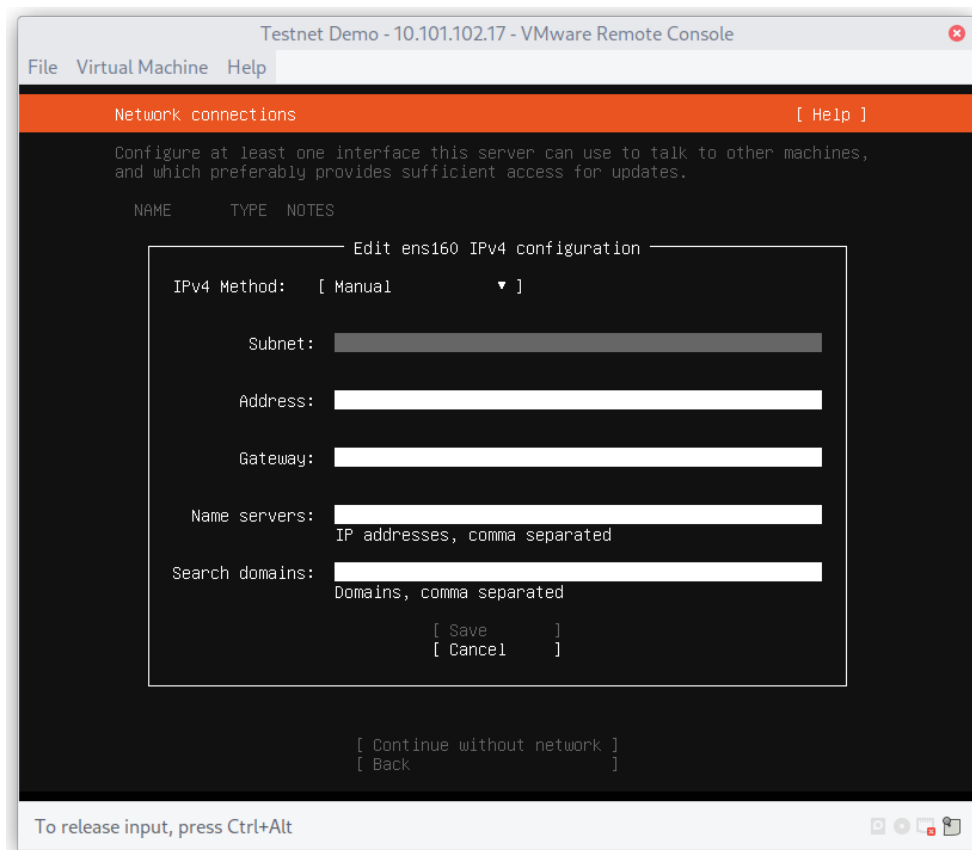
เลือกติดตั้ง Ubuntu Server



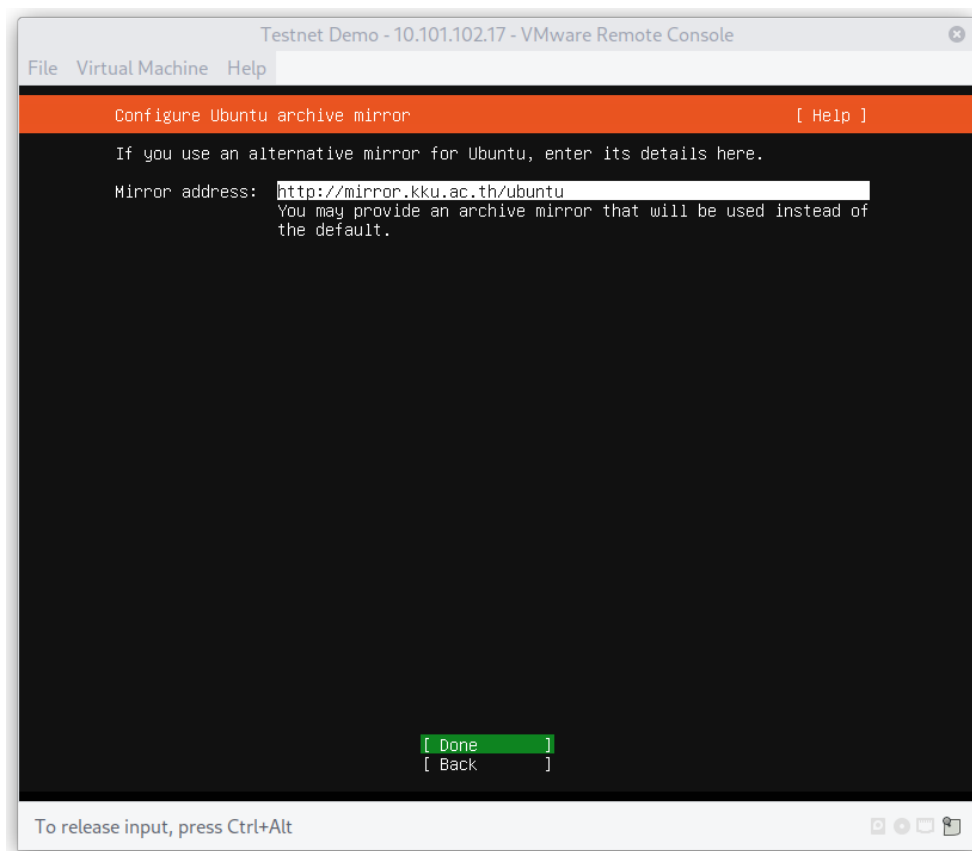
ตั้งค่าภาษาเป็น English



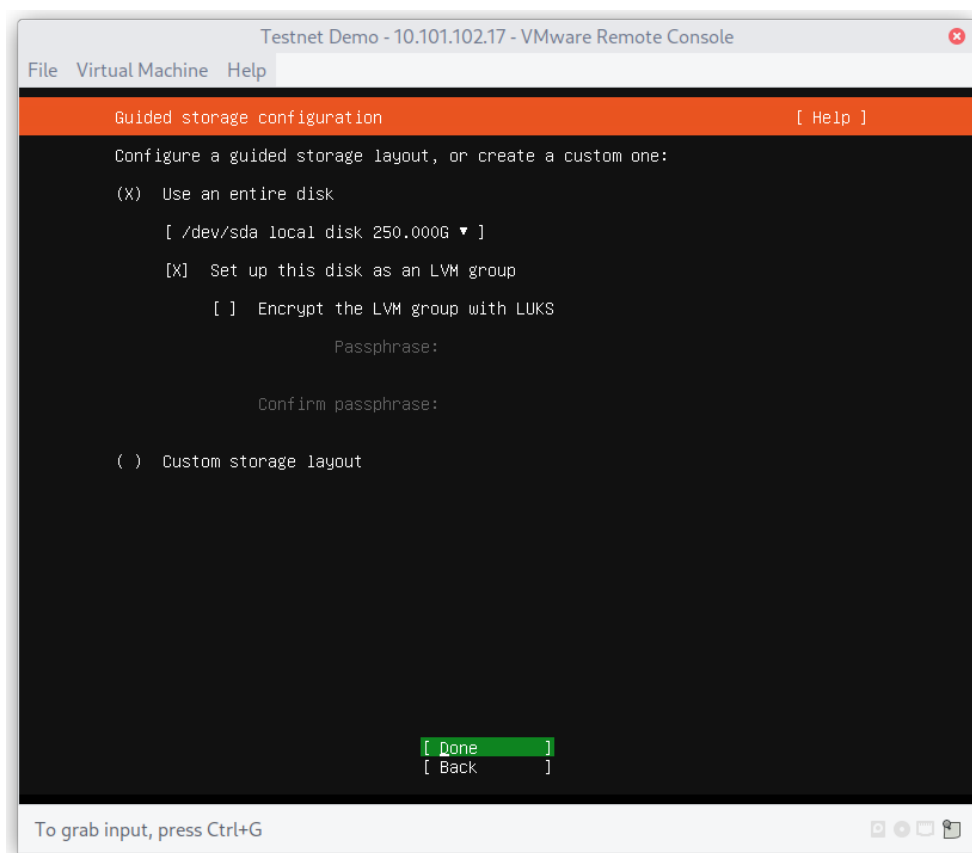
เลือก Ubuntu Server (minimized)



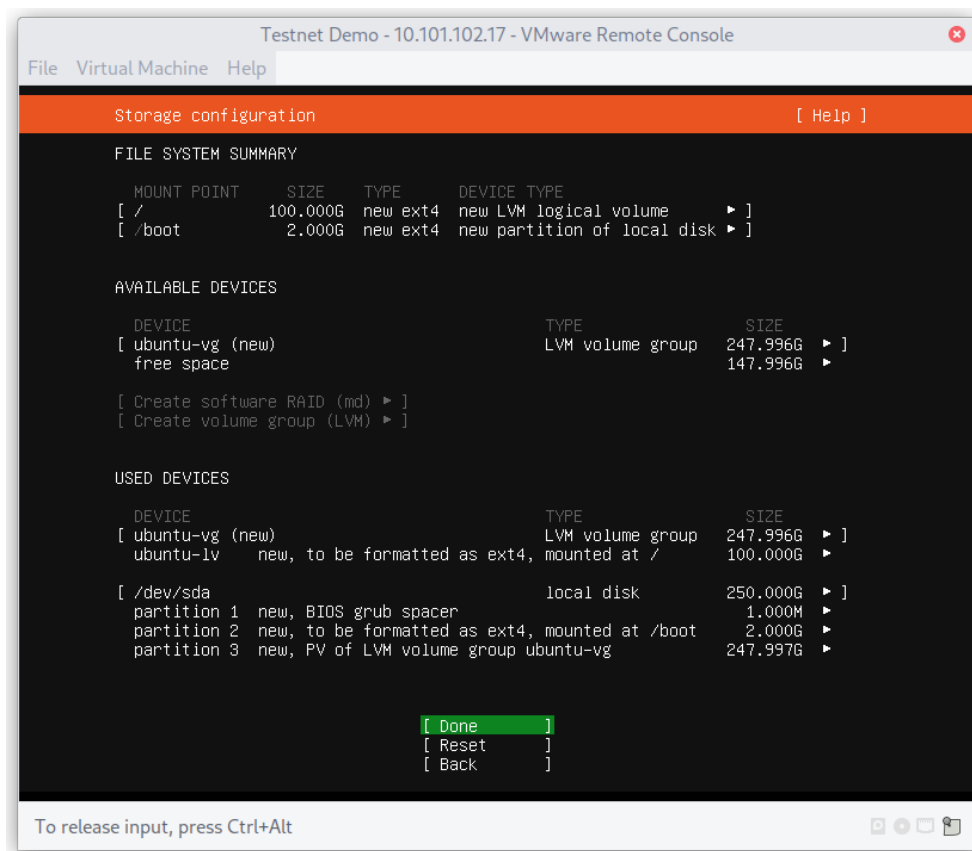
ตั้ง static IP Address



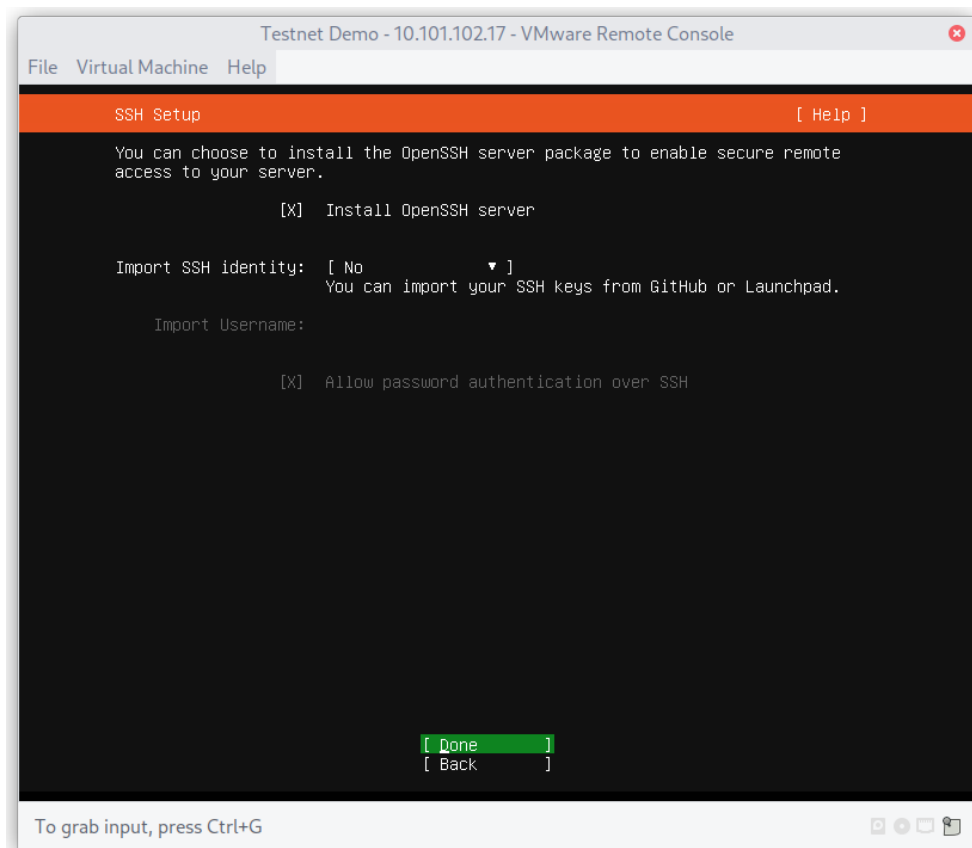
(optional) ตั้ง mirror เป็น <https://mirror.kku.ac.th/ubuntu>



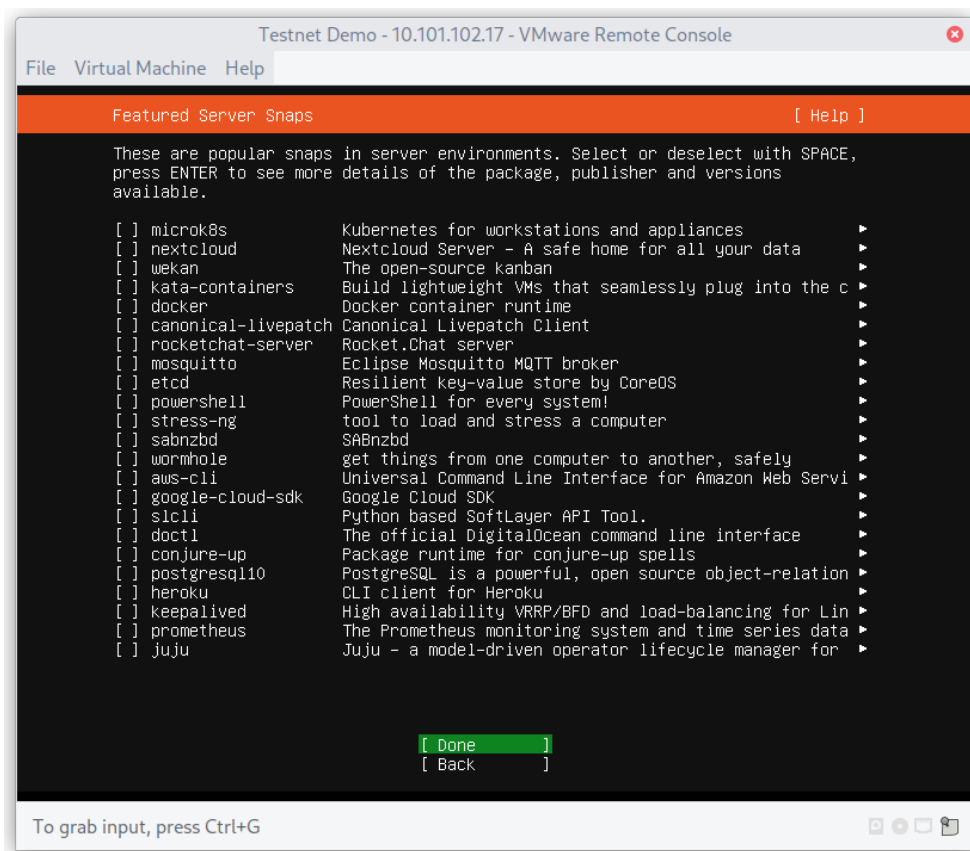
เลือก config เป็น LVM / entire disk



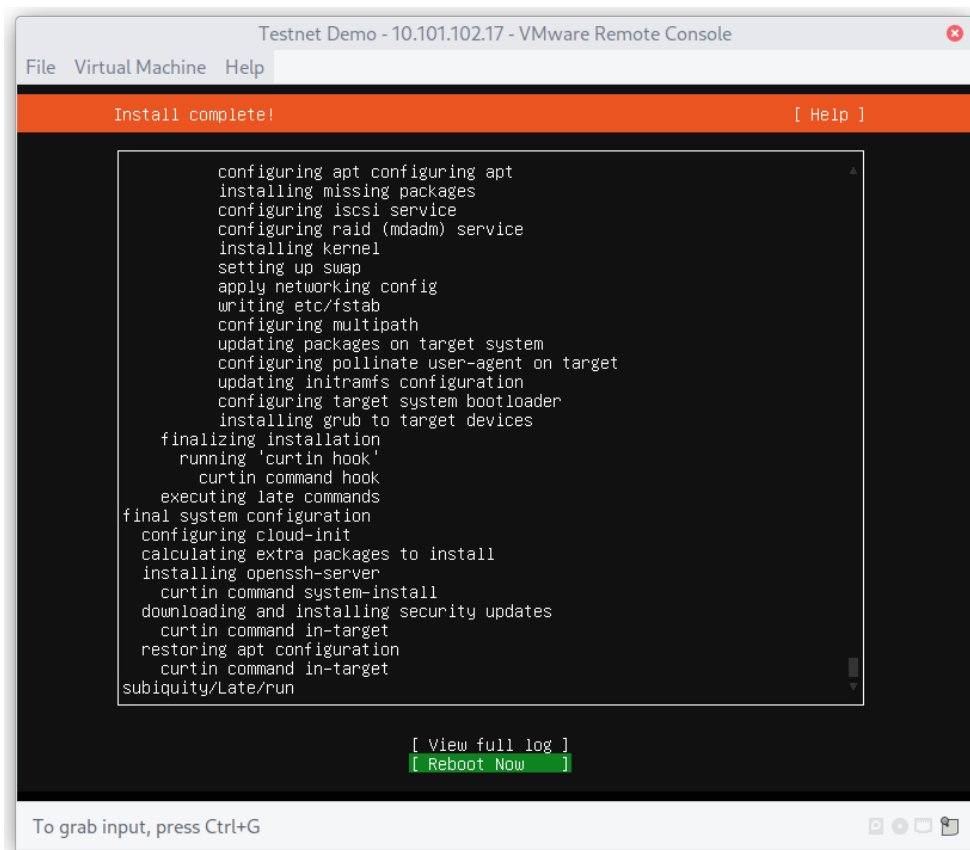
ยืนยัน storage configuration



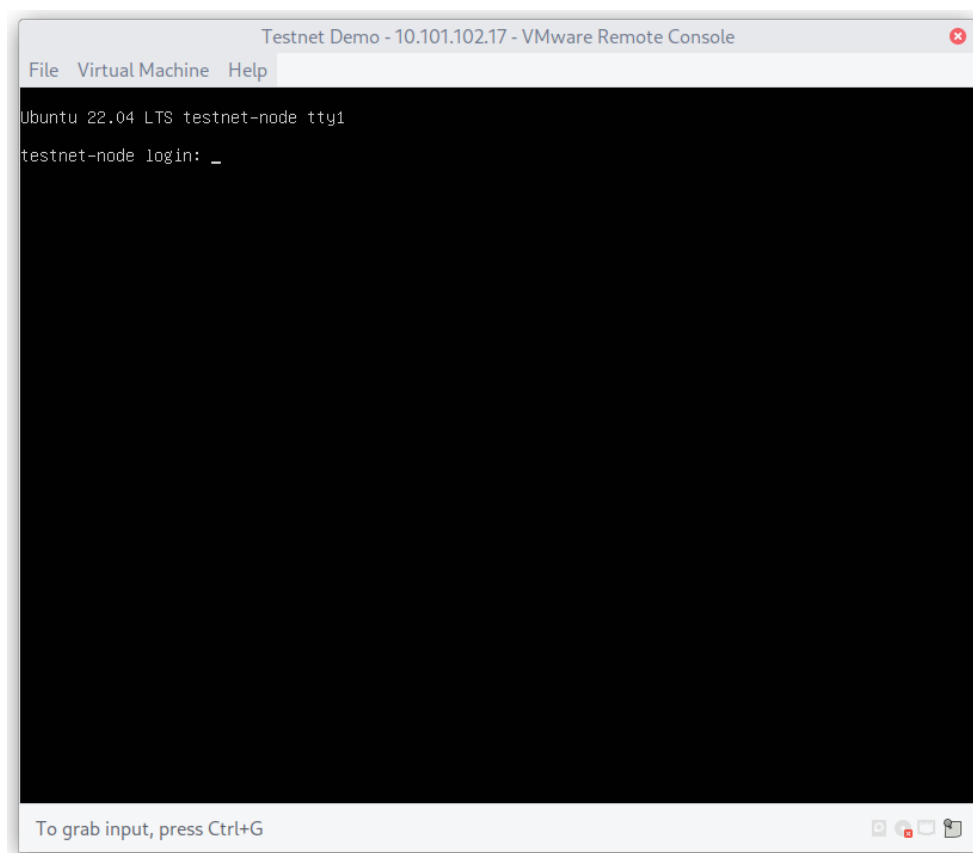
เลือกติดตั้ง OpenSSH server



ข้ามการติดตั้ง packages อื่น ๆ



รอดำเนินงาน complete / Reboot Now



หน้าจอ first boot

3.2. OpenSSH

3.2.1. Config OpenSSH ให้ใช้เฉพาะ Public Key Authenticaion

ที่ local machine, สร้าง key pair สำหรับ SSH และ copy public key ไปที่ signer :

```
$ ssh-keygen -t ed25519  
$ ssh-copy-id <signer hostname>
```

ที่ signer, แก้ config ให้ใช้เฉพาะ public-key authentication:

```
$ sudo vi /etc/ssh/sshd_config
```

แก้ไข:

```
PasswordAuthentication no  
ChallengeResponseAuthentication no  
KbdInteractiveAuthentication no
```

Restart ssh

```
$ sudo systemctl restart sshd
```

3.3. sudo

เพิ่ม non-root admin user เข้า group sudo เพื่อให้ใช้สิทธิ root โดยคำสั่ง sudo ได้

```
$ usermod -aG sudo <user>
```

ปิด root login

```
$ sudo passwd -l root
```

3.4. Set hostname

ตั้ง hostname (แนะนำเป็นชื่อ testnet-x หรือ mainnet-x)

```
$ sudo hostnamectl set-hostname <hostname>
```

re-login เพื่อยืนยันว่าเปลี่ยน hostname เรียบร้อยแล้ว

3.5. Set timezone and NTP

Set timezone

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

Set NTP server

```
$ sudo vi /etc/systemd/timesyncd.conf
```

Edit timesyncd.conf

```
NTP=<NTP Pool / Server>
```

Set NTP server

```
$ sudo timedatectl set-ntp off  
$ sudo timedatectl set-ntp on
```

สั่ง timedatectl เพื่อตรวจสอบ configuration

3.6. (Optional) ติดตั้ง NTP แทน timesyncd

Install NTP

```
$ sudo apt update  
$ sudo apt install ntp
```

3.7. (Optional สำหรับ VMware Guest) ติดตั้ง open-vm-tools

```
$ sudo apt install open-vm-tools
```

3.8. (Optional) ถอน snap ออกจากระบบ

```
$ sudo snap remove lxd  
$ sudo snap remove core20
```

```
$ sudo snap remove snapd
$ sudo apt purge snapd
```

4. ติดตั้ง Docker CE

4.1. ดำเนินการตามวิธีการในเพจ <https://docs.docker.com/engine/install/>

4.2. เพิ่ม user เข้า group docker เพื่อให้สามารถควบคุม docker container ได้โดยไม่ต้องใช้สิทธิ์ root

Add admin user to docker group

```
S sudo usermod -aG docker <user>
```

Login เข้า docker group

```
S newgrp docker
```

หรือ re-login ใหม่เพื่อ apply group

5. สร้าง Go Ethereum Container

5.1. สร้าง persistent volume สำหรับ node เช่น

Add admin user to docker group

```
$ sudo mkdir -p /srv/signer
$ sudo chown -R <user>:<group> /srv/signer
$ cd /srv/signer
```

configuration files จะเก็บไว้ที่ /srv/signer
node data จะเก็บไว้ที่ /srv/signer/node

5.2. Clone repository

For testnet validator node

```
$ apt install git
$ cd /srv/signer
$ git clone https://github.com/thai-blockchain/testnet.git
$ cd testnet
```

For mainnet validator node

```
$ apt install git
$ cd /srv/signer
$ git clone https://github.com/thai-blockchain/mainnet.git
$ cd mainnet
```

5.3. สร้างไฟล์รหัสผ่านสำหรับปลด private key

Create password.txt

```
$ sudo apt install pwgen
$ pwgen -Byn 32 1 > password.txt
$ chmod 400 password.txt
```

5.4. สร้าง wallet

```
$ docker run --rm -it -v $PWD:/srv/signer -w /srv/signer
  ethereum/client-go:stable --datadir /srv/signer/node --password
  password.txt account new
```

go-ethereum จะสร้าง keystore เก็บไว้ ประกอบด้วยไฟล์เก็บ private key ที่สามารถปลดล็อกเพื่อใช้งานโดยรหัสผ่านในไฟล์ password.txt

Wallet จะอ้างอิงจาก public address บันทึก public address ไว้ใช้งานต่อ และอ่านคำแนะนำ

```
You can share your public address with anyone. Others need it to interact with you.
You must NEVER share the secret key with anyone! The key controls access to your funds!
You must BACKUP your key file! Without the key, it's impossible to access account funds!
You must REMEMBER your password! Without the password, it's impossible to decrypt the
key!
```

5.5. Initialize node จาก genesis.json

```
$ docker run --rm -it -v $PWD:/srv/signer -w /srv/signer
  ethereum/client-go:stable --datadir /srv/signer/node init genesis.json
```

5.6. แก้ไข start-node.sh script, เปลี่ยน node name เป็นชื่อ node และ wallet เป็น public address:

```
--name <node name>
--unlock <wallet>
```

5.7. Start signer/validator container

```
$ ./start-node.sh
```

5.8. Monitor ได้โดยคำสั่ง

```
$ docker logs -f <container name>
$ docker logs -f --tail 10 <container name>
```

6. (Optional) ตั้งค่า firewall กรณีที่ต้องการต้องการทำ Access Control List ที่ validator nodes โดยใช้คำสั่ง ufw

6.1. ตั้ง default policies

```
$ sudo ufw default deny incoming
$ sudo ufw default allow outgoing
```

6.2. เพิ่ม rules สำหรับ ssh

```
$ sudo ufw allow ssh
```

6.3. อนุญาตการเชื่อมต่อจาก validator node แต่ละ node มาที่พอร์ต 30303/tcp รายการ IP addresses ของ validator node อยู่ในไฟล์ static-nodes.json และ Node Information

```
$ sudo ufw allow from <peer-1 IP address> to any port 30303
$ sudo ufw allow from <peer-2 IP address> to any port 30303
...
```

6.4. สั่ง ufw เริ่มทำงาน

คำเตือน: สั่ง ufw ทำงานอาจตัดการเชื่อมต่อที่มีอยู่ได้ แนะนำให้สั่งงานที่ console หรือ remote console ของ VM/VPS

```
$ sudo ufw enable
```

ตรวจสอบสถานะการทำงานของ ufw ได้ด้วยคำสั่ง

```
$ sudo ufw status
```

ตรวจสอบการทำงานว่าปกติหรือไม่ โดยดูจาก `admin.peers`, `clique.status()`, `docker logs validator` ควรสลับกัน sign block และ import new blocks จาก peer หากพบการทำงานผิดปกติ เช่น validator sign ทุก block หรือหยุด sign ให้ปิดการทำงานของ ufw ได้ด้วยคำสั่ง

```
$ sudo ufw disable
```

หากต้องการ clear rules ทั้งหมด ใช้คำสั่ง

```
$ sudo ufw reset
```

7. Basic Operations

เข้าใช้งาน web3 js console

```
$ ./attach.sh
```

แสดงจำนวน peer ที่เชื่อมต่ออยู่ขณะนั้น (Peer count)

```
> net.peerCount
```

แสดง peer ที่เชื่อมต่ออยู่ขณะนั้น

```
> admin.peers
```

เพิ่ม/เชื่อมต่อ peer ใหม่

```
> admin.addPeer("<enode@IP address:port>")
```

Promote/propose peer เข้ามาเป็น signer

```
> clique.propose("<wallet>", true)
```

Demote/propose peer ออกจากการเป็น signer

```
> clique.propose("<wallet>", false)
```

ตรวจสอบสถานะการ sign ของ Clique Protocol

```
> clique.status()
```

ลบ ledger และสถานะทั้งหมด

```
$ docker run --rm -it -v $PWD:/srv -w /srv ethereum/client-go:stable  
--datadir /srv/node removedb
```

Changelogs

Version	Date	Changes / Comments
1.2	2022-10-20	Update document to support mainnet installation
1.1	2022-10-15	Add (optional) firewall instruction
1.0	2022-07-17	Request for comments