🡪 **step-by-step guide for setting up a simple networked project through a command line interface.**

1. Set up **two VMs** using non-graphical installation images for the Linux distribution of your choice.

* Just download the ISO file and make an installation USB disk and install it on your computer / virtual machine offline. Once the ubuntu installation process is completed then we should create a clone of our original Linux ubuntu before we power on.

1. On one VM, configure and run an **SSH daemon** (e.g OpenSSH, dropbear, wolfSSH) that

**is reachable from the host machine**.

* First, we have to write **sudo apt install openssh-server** to install ssh-server, then we have to write **sudo systemctl enable ssh** to enable ssh. Then we have to write **sudo systemctl start ssh** to start the process.

**listens a non-default port.**

* In order to listen to a non-default port first of all we will have to write **sudo nano /etc/ssh/sshd\_config** to open the file to change the port. **22 = 2233, which was changed by me.**

**allows only unpriviliged users (non-root) to connect.**

* To do this we will have to the config mode by using **sudo nano /etc/ssh/sshd\_config.** Ones we are in the config mode then we will have to change the **permit root login = no.**

**disables password authentication in favour of public key authentication.**

* In order to do this we will have to go again in the config mode by using **sudo nano /etc/ssh/sshd\_config.** Once we are in the config mode, we will have to do multiple things. First of all, we should do **ChallengeResponseAuthentication = no** then **PasswordAuthentication = no** and at the last **UsePAM = no.**

**starts automatically upon boot.**

In order to do this, we should use **sudo systemctl enable ssh** to enable ssh and start up.

1. Configure an instance of a **service** (e.g. a web server or IRC) on each VM.

* In order to install the IRC, we have to write the command **sudo apt-get install hexchat**

1. Set up a **virtual private network** using **Wireguard** or **tinc** connecting the two VMs.

* In order to install **Wireguard** following steps should be followed but it is not necessary if you have it already.

In order to update the package or managing the system repositories these given commands should be used.

**Sudo apt update**

**Sudo apt install software-properties-common**

But probably your system is already up to date.

* In order to add the wireguard repositories this given command must be used.

**Sudo add-apt-repository ppa:wireguard/wireguard.**

* To install wireguard this given command should be written in terminal.

**Sudo apt install wireguard.**

* Once the installation process is completed the next step is to be configuring the **Wireguard.**

First of all, a public key as well as a private key should be generated. By using this given command these keys can be generated.

**wg genkey | sudo tee /etc/wireguard/privatekey | wg pubkey | sudo tee /etc/wireguard/publickey.**

* The next step is to create a new file name, in order to add to add following things.

**[wg0]**

**Address = 10.0.0.1/24**

**SaveConfig = true**

**ListenPort = 51820**

**PrivateKey = SERVER\_PRIVATE\_KEY**

**PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE**

**PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -o ens3 -j MASQUERADE**

* After this step we will have to write a command to check the status and configuration.

**Sudo wg show wg0**

* To bring the **Wireguard** interface at boot time run the following command:

**Sudo systemctl enable wg-quick@wg0**

* The next step is to the process for setting up a Linux. We start by generating public and private key.

**wg genkey | sudo tee /etc/wireguard/privatekey | wg pubkey | sudo tee /etc/wireguard/publickey**

* The next step is to create a file wg0.conf and add.

**Sudo nano /etc/wireguard/wg0.conf**

Once you are in the configure mode the following things should be write.

[Iwg0]

PrivateKey = CLIENT\_PRIVATE\_KEY

Address = 0.0.0.0/0

[wg0]

PublicKey = SERVER\_PUBLIC\_KEY

Endpoint = SERVER\_IP\_ADDRESS: ----

AllowedIPs = 0.0.0.0/0

1. Configure a firewall through ufw to restrict access to ports to a minimum.

* First of all, we will have to install ufw, if its is not installed in our ubuntu.
* In order to check whether ufw is already installed or not, following commands should be used.

**sudo dpjk - -get-selection | grep ufw**

if ufw is not installed then by using this following command it can be installed **sudo apt-get install ufw** after that the status should be checked, whether ufw is active or not. In order to check the status this given command should be written **sudo ufw status** after that you can add rules.

Before you add new rules if you want to check/see the default rules then by using this command **sudo ufw allow ssh** then you can write **sudo ufw allow ssh/tcp** then you can add a port number by using **sudo ufw allow 2233/tcp**

* SSH access from the host machine to the first VM.
* First of all we will have to check whether SSH is already running in our ubuntu if not then by using this given command **sudo service ssh start**

it can be started.

* In order to login to linux virtual machine by using SSH first of all some settings should be changed on virtual box. To change the settings follow the given steps.

**Setting -> Network -> Advance -> port forwarding** after that you have to add some important things like. Rule, TCP, Host IP, Host port, Guest IP, Guest port. if you want to see, what your guest IP is then by using this command **Ifconfig eth0** you can see it then you can add it. After that by using this command **ssh** [**username@127.0.0.1**](mailto:username@127.0.0.1) **-p 2233** you can login to your linux virtual machine.

**Username = name of your linux virtual machine.**