Ubuntu 20.04.3 server setup guide

Guides Followed:

- https://hibbard.eu/install-ubuntu-virtual-box/
- https://linuxconfig.org/how-to-install-and-configure-dropbear-on-linux

Pre install task

- install Ubuntu 20.04.3 to your computer from: https://ubuntu.com/download/server
- install VM to your computer from: https://www.virtualbox.org/wiki/Downloads

VM setup

- Name your server and set up a operating system
-
  - Server name : u02 and for operating system i chose linux with Ubuntu (64bit)
- Chose a memory size and create a virtual hard disk
-
  - I recommend using the recommended memory size as it is
-
  - Create a new harddisk
- ------- VDI (Virtualbox Disk Image)
- ------- Dynamically allocated
- ------- File location and size: Locate to the ubuntu folder and set the size to 10 gb

Dropbear and firewall (ufw) setup

- Make sure that you're signed as root
-
  - root@laptop:/home/semha# apt install dropbear Reading package lists... Done Building dependency tree Reading state information... Done dropbear is already the newest version (2019.78-2build1). 0 upgraded, 0 newly installed, 0 to remove and 42 not upgraded.

-- After dropbear is installed we need to set it up using nano -- "semha@laptop:~$ nano /etc/default/dropbear" -- Set No_start to 0 and Dropbear_port to the port you wanna use (i use 35001)

Dropbear

-# change to NO_START=0 to enable Dropbear -NO_START=0 -# the TCP port that Dropbear listens on DROPBEAR_PORT=35001

-# any additional arguments for Dropbear -DROPBEAR_E TRA_ARGS=

-# specify an optional banner file containing a message to be -# sent to clients before they connect, such as "/etc/issue.net" -DROPBEAR_BANNER=""

-# RSA hostkey file (default: /etc/dropbear/dropbear_rsa_host_key) -#DROPBEAR_RSAKEY="/etc/dropbear/dropbear_rsa_host_key"

-# DSS hostkey file (default: /etc/dropbear/dropbear_dss_host_key) - #DROPBEAR_DSSKEY="/etc/dropbear/dropbear_dss_host_key"

-# ECDSA hostkey file (default: /etc/dropbear/dropbear_ecdsa_host_key) - #DROPBEAR_ECDSAKEY="/etc/dropbear/dropbear_ecdsa_host_key

- Restart your dropbear with systemctl restart dropbear
- Before you try to SSH in to your server make sure your firewall is setup and you do it by following these steps

-- root@laptop:/home/semha# sudo apt install ufw Reading package lists... Done Building dependency tree Reading state information... Done The following packages will be upgraded: ufw 1 upgraded, 0 newly installed, 0 to remove and 42 not upgraded.

The next step is to set up the rules that ufw will follow -- root@laptop:/home/semha# sudo ufw allow 35001 Rule updated Rule updated (v6) --And then just reboot root@laptop:/home/semha# reboot root@laptop:/home/semha# Session terminated, killing shell...exit ...killed. Terminated

-- Make sure that it is enabled and active semha@laptop:~$ sudo ufw enable [sudo] password for semha: Firewall is active and enabled on system startup

root@laptop:/home/semha# ufw status verbose Status: active Logging: on (low) Default: deny (incoming), allow (outgoing), disabled (routed) New profiles: skip

To Action From

35001 ALLOW IN Anywhere 35001 (v6) ALLOW IN Anywhere (v6)

Set up your SSH keys

- First step is to set it up on your own local computer
- - haail@LAPTOP-SP9U89DI MINGW64 ~ (main) $ ssh-keygen Generating public/private rsa key pair. Enter file in which to save the key (/c/Users/haail/.ssh/id_rsa): /c/Users/haail/.ssh/id_rsa already exists. Overwrite (y/n)? y Enter passphrase (empty for no passphrase): Enter same passphrase again: Your identification has been saved in /c/Users/haail/.ssh/id_rsa Your public key has been saved in /c/Users/haail/.ssh/id_rsa.pub The key fingerprint is:
- Now we have to make a copy on our local computer haail@LAPTOP-SP9U89DI MINGW64 ~ (main) $ cat ~/.ssh/id_rsa.pub | ssh -p 35001 semha@172.20.10.3 "mkdir -p 35001 ~/.ssh && touch ~/.ssh/authorized_keys" semha@172.20.10.3's password:
- Create a directory to store your keys haail@LAPTOP-SP9U89DI MINGW64 ~ (main) $ mkdir -p ~/.ssh
- Move your keys to the directory haail@LAPTOP-SP9U89DI MINGW64 ~ (main) $ echo public_key_string >> ~/.ssh/authorized_keys
- Remove Group and others permission haail@LAPTOP-SP9U89DI MINGW64 ~ (main) $ chmod -R go= ~/.ssh

Now its time to set it up on your own VM

semha@laptop:~$ ssh-keygen Generating public/private rsa key pair. Enter file in which to save the key (/home/semha/.ssh/id_rsa): /home/semha/.ssh/id_rsa already exists. Overwrite (y/n)? y Enter passphrase (empty for no passphrase): Enter same passphrase again: Your identification has been saved in /home/semha/.ssh/id_rsa Your public key has been saved in /home/semha/.ssh/id_rsa.pub The key fingerprint is:

- Make a copy of the key haail@LAPTOP-SP9U89DI MINGW64 ~ (main) $ ssh -p 35001 semha@172.20.10.3 semha@172.20.10.3's password: semha@laptop:~$ ssh-copy-id -p 35001 semha@172.20.10.3 /usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/semha/.ssh/id_rsa.pub" The authenticity of host '[172.20.10.3]:35001 ([172.20.10.3]:35001)' can't be established. ECDSA key fingerprint is SHA0000000000000000000000000000000000000000000000000000. Are you sure you want to continue connecting (yes/no/[fingerprint])? y Please type 'yes', 'no' or the 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 semha@172.20.10.3's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh -p '35001' 'semha@172.20.10.3'" and check to make sure that only the key(s) you wanted were added.