How to connect Ubuntu to AD

Made: 12/22/21

Version: Ubuntu 20.04 | Win10

Author: Aaron Sierra

NOTE: anything with a box is a command you will type into the terminal

---

Steps

□ sudo su (you need elevated privileges for a lot of the stuff here.)

□ apt update

□ apt upgrade

NOTE: your machine will experience segmentation fault if you fail to do above commands.

- Install needed packages
- Configure files in packages
- Join AD with realmd
- config files a little more
- Verify you joined Active Directory successfully

First we are going to install needed packages

Below are the packages you need to install, you can do them individually or do them in bulk

□ apt install realmd

□ apt install libnss-sss

□ apt install libpam-sss

□ apt install sssd

□ apt install sssd-tools

□ apt install adcli

□ apt install samba-common-bin

□ apt install oddjob

□ apt install oddjob-mkhomedir

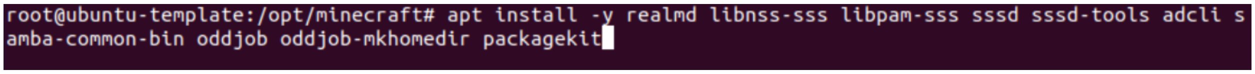
□ apt install package kit

□ apt install ssh

In Case you want to do it in one line

□ apt install -y ssh realmd libnss-sss libpam-sss sssd sssd-tools adcli samba-common-bin oddjob oddjob-mkhomedir packagekit

NOTE: this image does not include ssh, and I recieve an error, at the end of the checklist, but a solution is shown with pictures, ***make sure you have ssh installed***



```
root@ubuntu-template:/opt/minecraft# apt install -y realmd libnss-sss libpam-sss sssd sssd-tools adcli s
amba-common-bin oddjob oddjob-mkhomedir packagekit
```

Here is an image showing the command for all in one line.

Next is the hostname

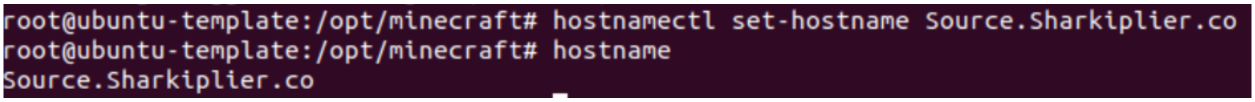
Host name is the name of the computer and it is preferable for the hostname to match the name of the AD you are connecting to.

□ hostnamectl set-hostname {name of the AD}

Here hostnamectl is the type of command you are going to run, similar to apt for apt install. The set-hostname, changes the name of you computer so that when you join the AD you know what computer has joined it.

□ hostname

typing this afterwards simply verifies that the command has worked.



```
root@ubuntu-template:/opt/minecraft# hostnamectl set-hostname Source.Sharkiplier.co
root@ubuntu-template:/opt/minecraft# hostname
Source.Sharkiplier.co
```

Image depicts the commands explained earlier.

Preparing Resolved Service

We are going to use the resolve service to connect our Ubuntu machine to the AD, however right now it is more than likely in use. We are going to disable it for now.

□ systemctl disable systemd-resolved.service

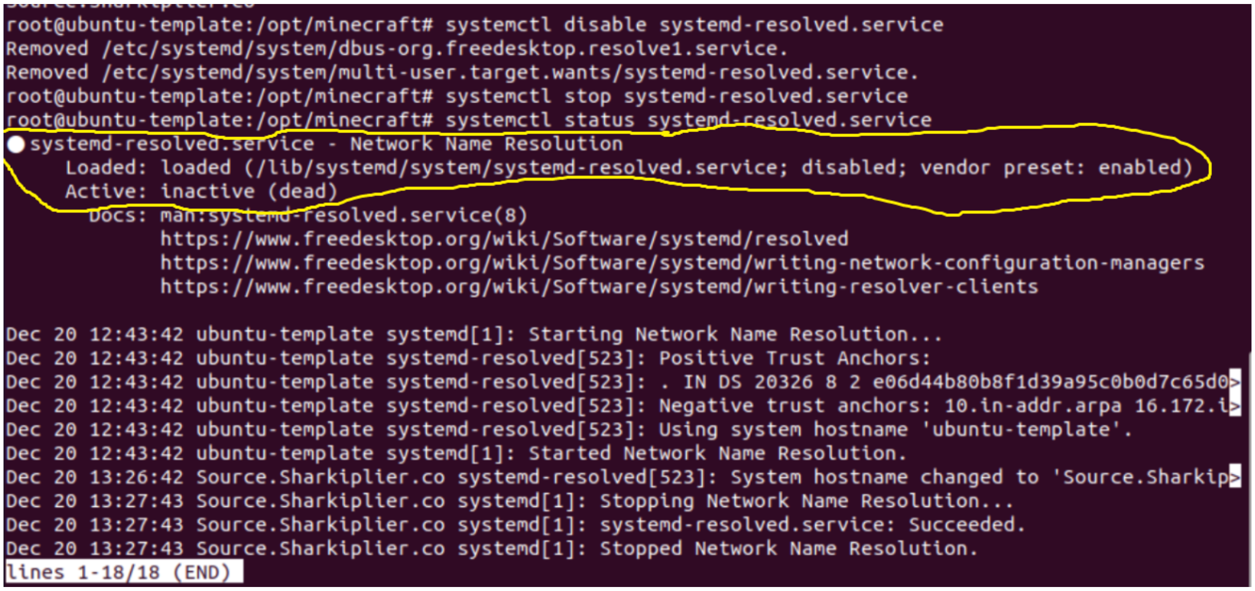
We disabled the service and now it should not be working, but it still may be running in the background.

□ systemctl stop systemd-resolved.service

We have killed its process in the background

□ systemctl status systemd-resolved.service

We check to see if our commands have successfully worked



```
root@ubuntu-template:/opt/minecraft# systemctl disable systemd-resolved.service
Removed /etc/systemd/system/dbus-org.freedesktop.resolve1.service.
Removed /etc/systemd/system/multi-user.target.wants/systemd-resolved.service.
root@ubuntu-template:/opt/minecraft# systemctl stop systemd-resolved.service
root@ubuntu-template:/opt/minecraft# systemctl status systemd-resolved.service
● systemd-resolved.service - Network Name Resolution
     Loaded: loaded (/lib/systemd/system/systemd-resolved.service; disabled; vendor preset: enabled)
     Active: inactive (dead)
       Docs: man:systemd-resolved.service(8)
             https://www.freedesktop.org/wiki/Software/systemd/resolved
             https://www.freedesktop.org/wiki/Software/systemd/writing-network-configuration-managers
             https://www.freedesktop.org/wiki/Software/systemd/writing-resolver-clients

Dec 20 12:43:42 ubuntu-template systemd[1]: Starting Network Name Resolution...
Dec 20 12:43:42 ubuntu-template systemd-resolved[523]: Positive Trust Anchors:
Dec 20 12:43:42 ubuntu-template systemd-resolved[523]: . IN DS 20326 8 2 e06d44b80b8f1d39a95c0b0d7c65d0>
Dec 20 12:43:42 ubuntu-template systemd-resolved[523]: Negative trust anchors: 10.in-addr.arpa 16.172.i>
Dec 20 12:43:42 ubuntu-template systemd-resolved[523]: Using system hostname 'ubuntu-template'.
Dec 20 12:43:42 ubuntu-template systemd[1]: Started Network Name Resolution.
Dec 20 13:26:42 Source.Sharkiplier.co systemd-resolved[523]: System hostname changed to 'Source.Sharkip>
Dec 20 13:27:43 Source.Sharkiplier.co systemd[1]: Stopping Network Name Resolution...
Dec 20 13:27:43 Source.Sharkiplier.co systemd[1]: systemd-resolved.service: Succeeded.
Dec 20 13:27:43 Source.Sharkiplier.co systemd[1]: Stopped Network Name Resolution.
lines 1-18/18 (END)
```

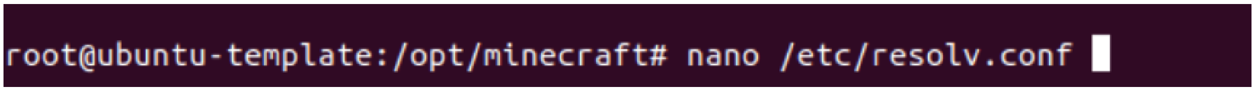
Here is an image of the commands we ran, and what is important here is the yellow circle, this shows up that the status has successfully been disabled and is inactive (dead)

Resolved config files

Now we are going to edit the resolved config files to make the nameserver match the AD’s ip address

- ☐ nano /etc/resolv.conf

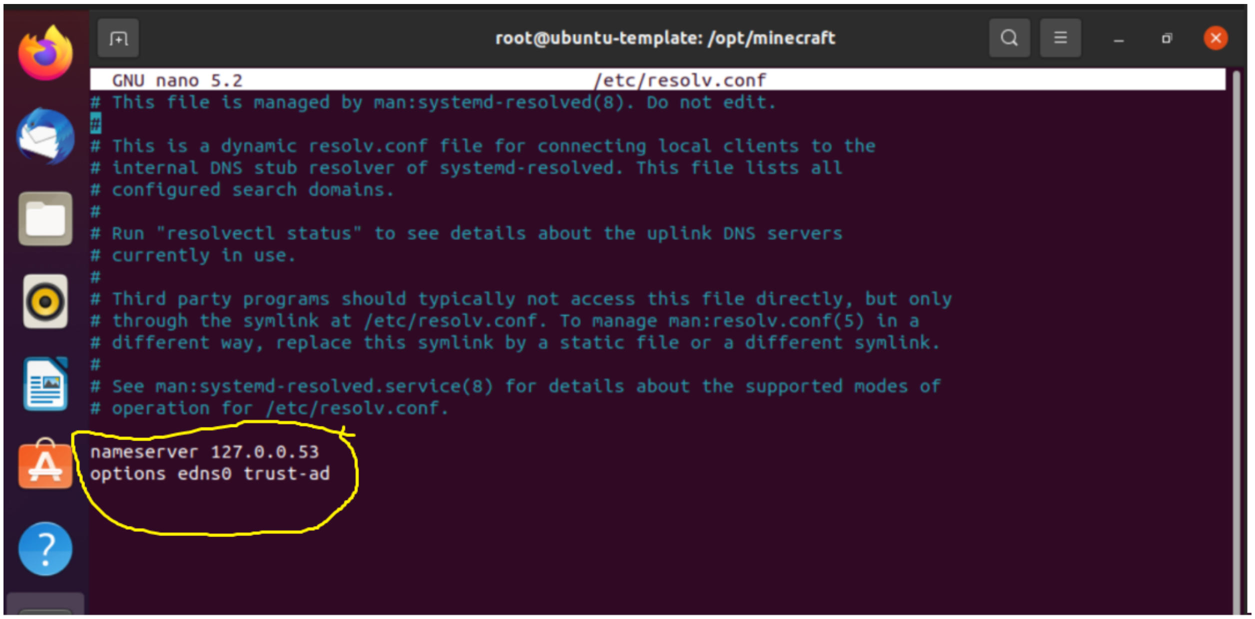
This command opens the resolv.conf document and allows you to edit it, simillaryly to a google document. Nano at the beginning of the command can be replaced with vi or another file editor you prefer.



```
root@ubuntu-template:/opt/minecraft# nano /etc/resolv.conf
```

The command

Now we change the namerserver (by default should be 127.0.0.#) to the AD

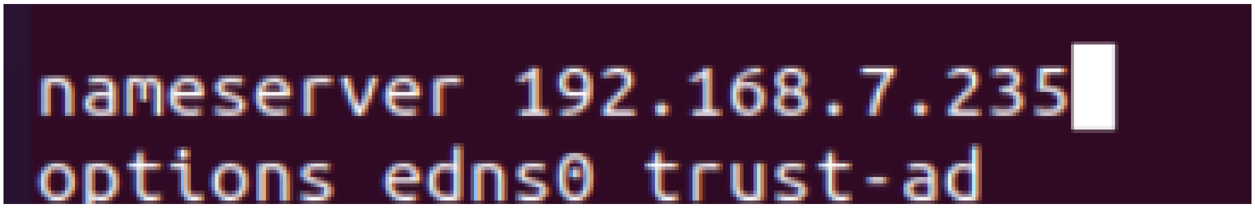


```
  GNU nano 5.2                          /etc/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
```

Here is an image depicting what the default conf file should look like, we are going to change what is circled in yellow

Change the nameserver to AD ip

In this case the AD ip was 192.168.7.235



```
nameserver 192.168.7.235
options edns0 trust-ad
```

The circled now has the IP of our specific AD

NOTE: in order to save changed you made to a file you must say yes to “save changes?”

Now we should be ready to join the AD

□ realm discover {your domain name of the AD} //(in this case it was Sharkiplier.co)

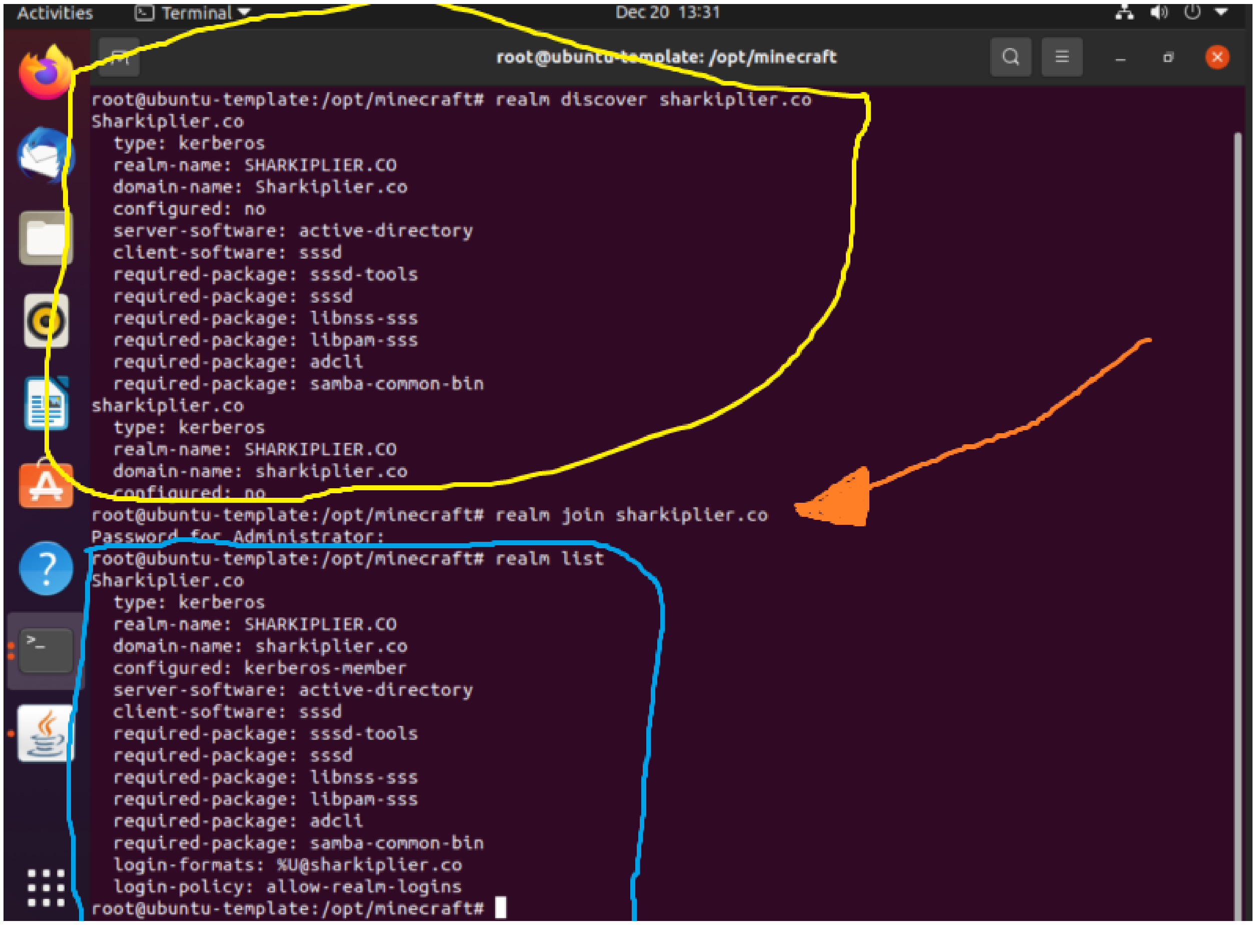
This command ensures you are able to find the domain of the AD

□ realm join {your domain name of the AD}

This command joins the domain of the AD after verifying that you have found it

- □ realm list

This shows a list of the realms you are connected to, so if you successfully joined the domain of the AD then it should appear here.



```
root@ubuntu-template:/opt/minecraft# realm discover sharkiplier.co
Sharkiplier.co
  type: kerberos
  realm-name: SHARKIPLIER.CO
  domain-name: Sharkiplier.co
  configured: no
  server-software: active-directory
  client-software: sssd
  required-package: sssd-tools
  required-package: sssd
  required-package: libnss-sss
  required-package: libpam-sss
  required-package: adcli
  required-package: samba-common-bin
sharkiplier.co
  type: kerberos
  realm-name: SHARKIPLIER.CO
  domain-name: sharkiplier.co
  configured: no
root@ubuntu-template:/opt/minecraft# realm join sharkiplier.co
Password for Administrator:
root@ubuntu-template:/opt/minecraft# realm list
Sharkiplier.co
  type: kerberos
  realm-name: SHARKIPLIER.CO
  domain-name: sharkiplier.co
  configured: kerberos-member
  server-software: active-directory
  client-software: sssd
  required-package: sssd-tools
  required-package: sssd
  required-package: libnss-sss
  required-package: libpam-sss
  required-package: adcli
  required-package: samba-common-bin
  login-formats: %U@sharkiplier.co
  login-policy: allow-realm-logins
root@ubuntu-template:/opt/minecraft#
```

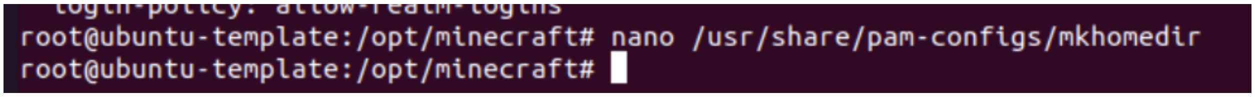
This is an image of the commands you will type and what they should return, the yellow is for the first command, orange for the second, and blue for the third.

Pam configs

Now that we have joined the AD its done right? No, if you want it to work properly there are a few more things you have to configure.

- □ nano /usr/share/pam-configs/mkhomedir

This opens a config file regarding home directories for users in the domain for you to edit.

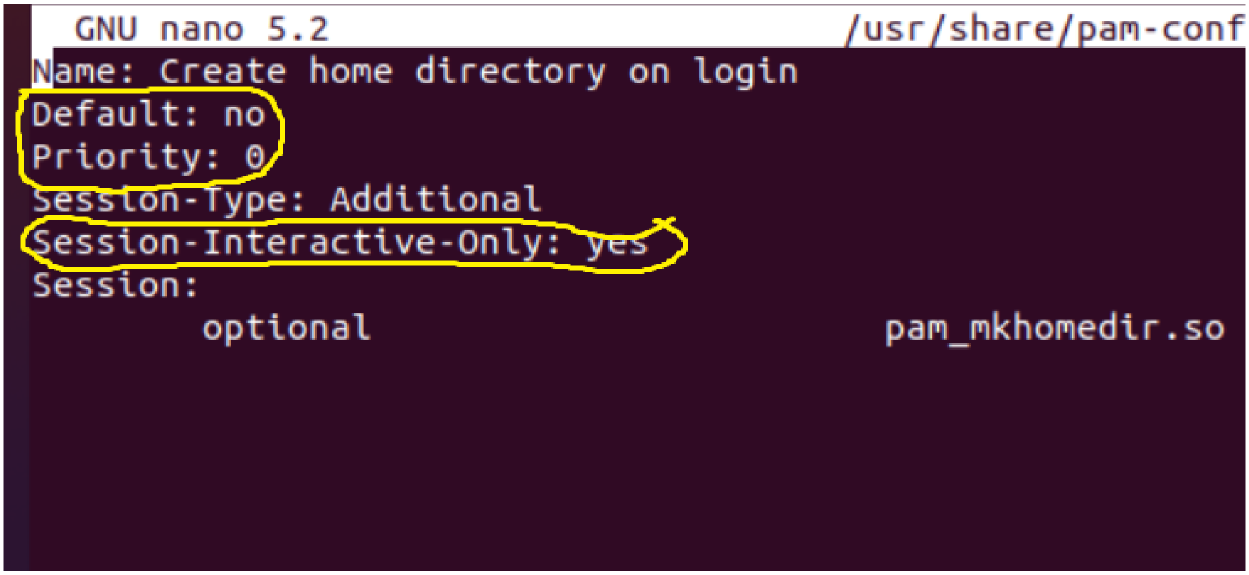


```
login-policy: allow-realm-logins
root@ubuntu-template:/opt/minecraft# nano /usr/share/pam-configs/mkhomedir
root@ubuntu-template:/opt/minecraft#
```

The command

Here is what the default file will look like

what is circled in yellow is what you are going to change

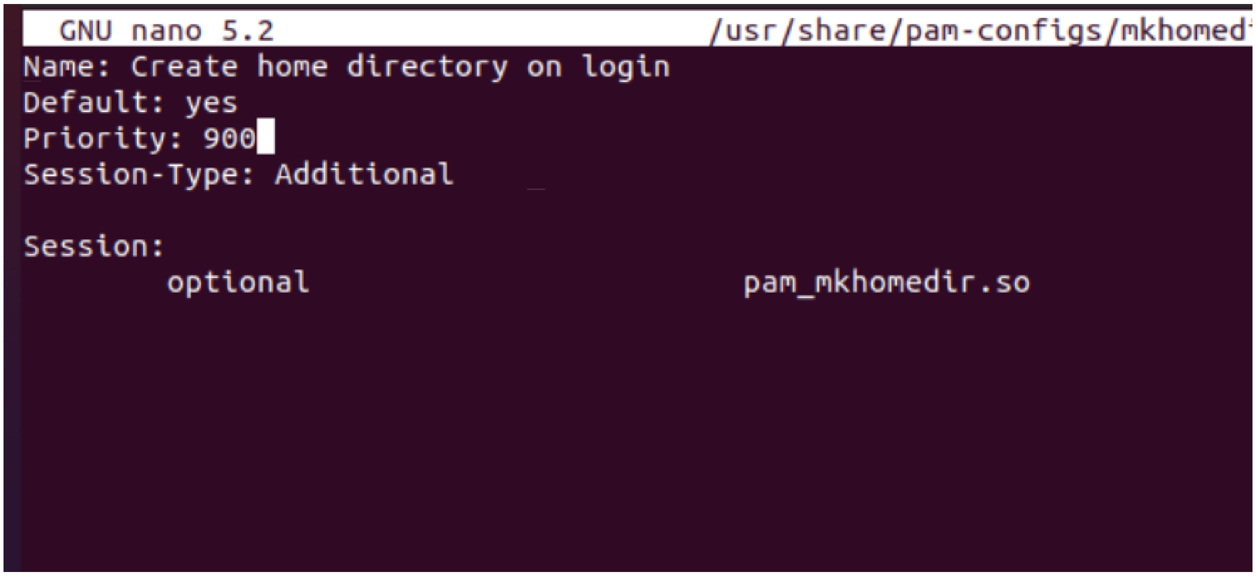


```
  GNU nano 5.2                                  /usr/share/pam-conf
Name: Create home directory on login
Default: no
Priority: 0
Session-Type: Additional
Session-Interactive-Only: yes
Session:
        optional                        pam_mkhomedir.so
```

We are going to change default to yes: (to make new home directories for users)

change Priority to 999

delete session-interactive only so we can have different types of sessions



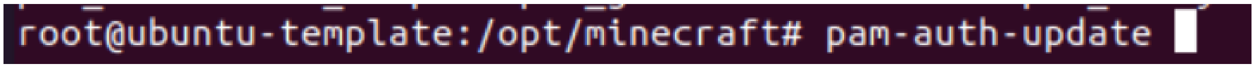
```
  GNU nano 5.2                       /usr/share/pam-configs/mkhomed
Name: Create home directory on login
Default: yes
Priority: 900
Session-Type: Additional

Session:
        optional                        pam_mkhomedir.so
```

This is what it should look like afterward

Now for Pam files

- □ pam-auth-update

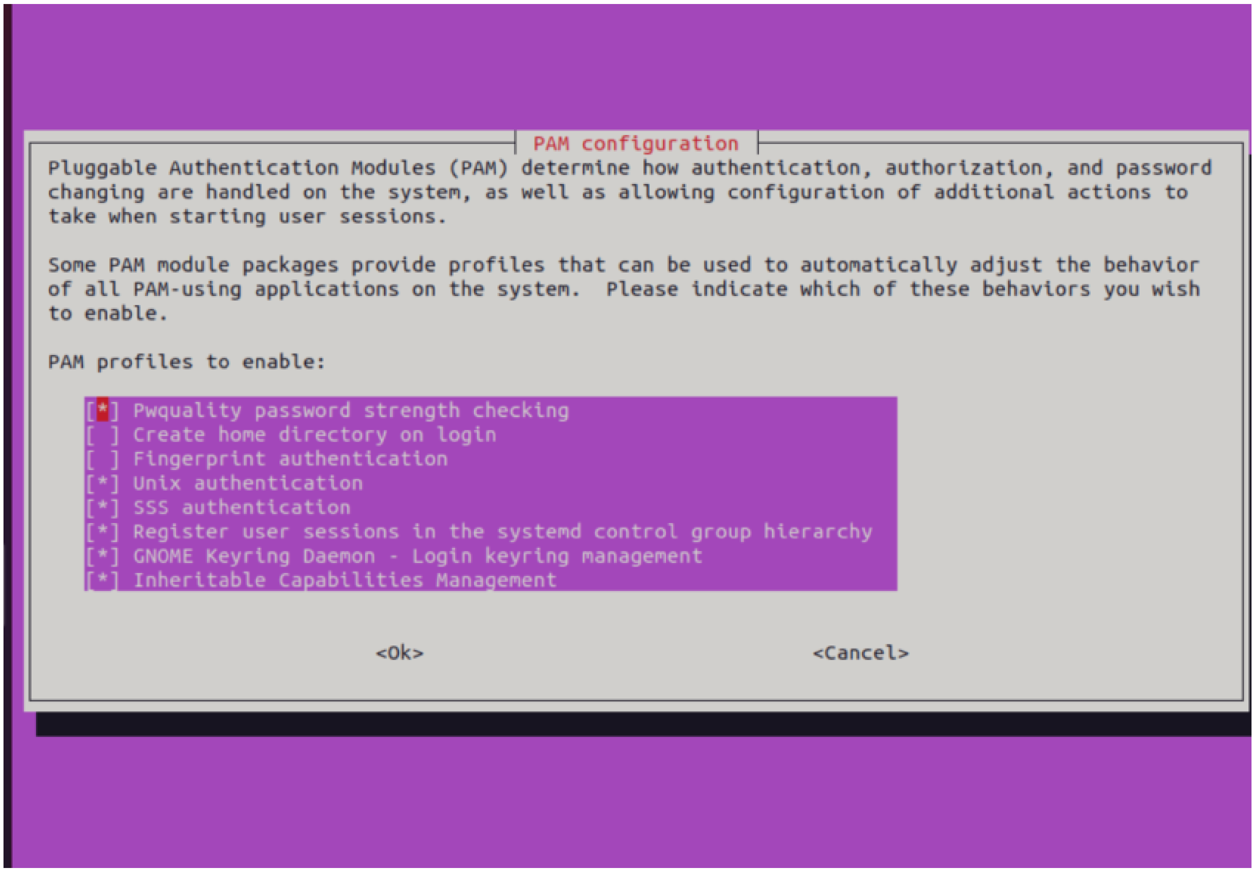
This opens pam config and you only have to change one thing



```
root@ubuntu-template:/opt/minecraft# pam-auth-update
```

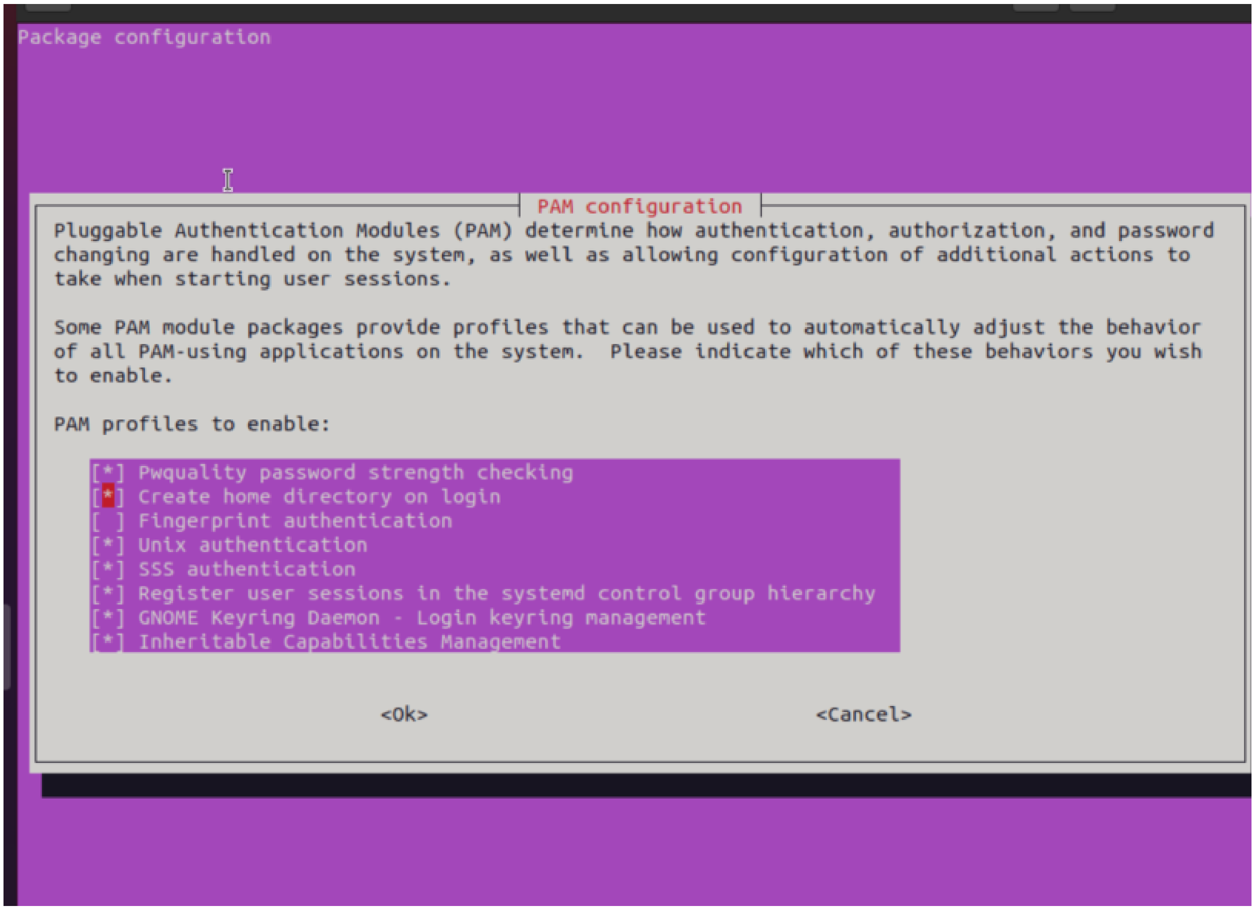
picture of the command

You will be met with a screen like this



```
PAM configuration

Pluggable Authentication Modules (PAM) determine how authentication, authorization, and password
changing are handled on the system, as well as allowing configuration of additional actions to
take when starting user sessions.

Some PAM module packages provide profiles that can be used to automatically adjust the behavior
of all PAM-using applications on the system.  Please indicate which of these behaviors you wish
to enable.

PAM profiles to enable:

   [*] Pwquality password strength checking
   [ ] Create home directory on login
   [ ] Fingerprint authentication
   [*] Unix authentication
   [*] SSS authentication
   [*] Register user sessions in the systemd control group hierarchy
   [*] GNOME Keyring Daemon - Login keyring management
   [*] Inheritable Capabilities Management


                    <Ok>                                    <Cancel>
```

using the arrow keys, navigate to the “Create home directory on login” and press spacebar



```
Package configuration

                          PAM configuration
 Pluggable Authentication Modules (PAM) determine how authentication, authorization, and password
 changing are handled on the system, as well as allowing configuration of additional actions to
 take when starting user sessions.

 Some PAM module packages provide profiles that can be used to automatically adjust the behavior
 of all PAM-using applications on the system.  Please indicate which of these behaviors you wish
 to enable.

 PAM profiles to enable:

    [*] Pwquality password strength checking
    [*] Create home directory on login
    [ ] Fingerprint authentication
    [*] Unix authentication
    [*] SSS authentication
    [*] Register user sessions in the systemd control group hierarchy
    [*] GNOME Keyring Daemon - Login keyring management
    [*] Inheritable Capabilities Management


                      <Ok>                                  <Cancel>
```

Hit enter and your done with this page

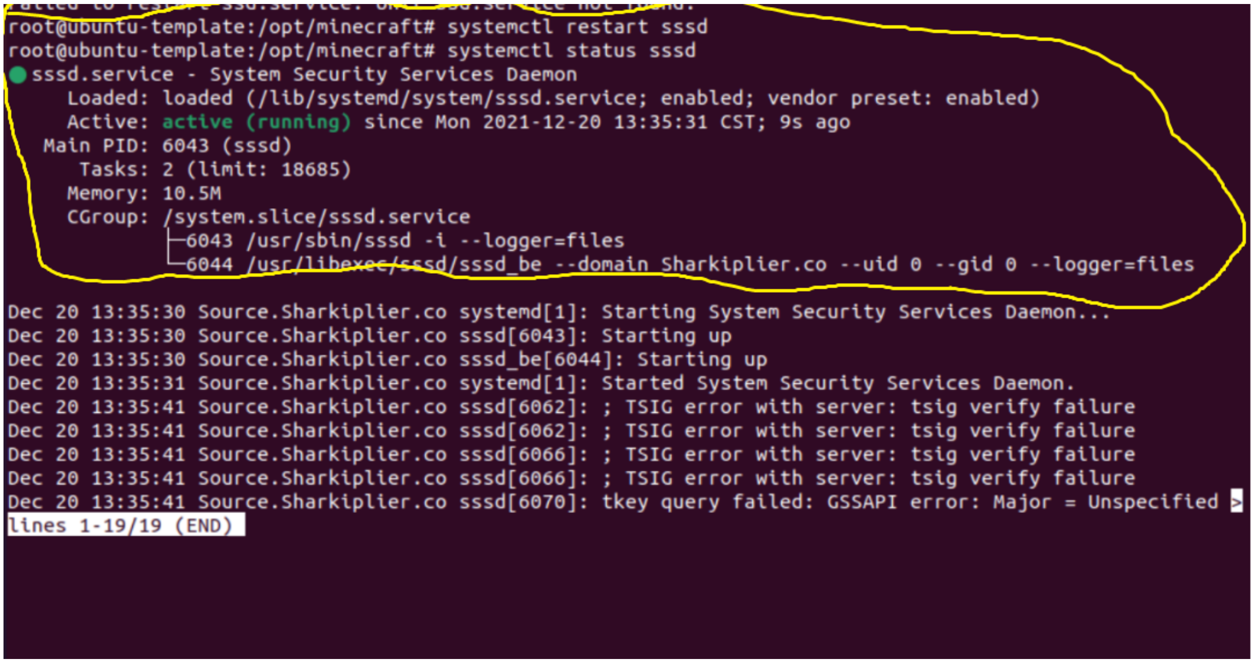
Rebooting SSSD

- □ systemctl restart sssd

This will restart it and enable it to work without computer restart

- □ systemctl status sssd

This shows the status of the SSSD to show that it is working



```
root@ubuntu-template:/opt/minecraft# systemctl restart sssd
root@ubuntu-template:/opt/minecraft# systemctl status sssd
● sssd.service - System Security Services Daemon
     Loaded: loaded (/lib/systemd/system/sssd.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2021-12-20 13:35:31 CST; 9s ago
   Main PID: 6043 (sssd)
      Tasks: 2 (limit: 18685)
     Memory: 10.5M
     CGroup: /system.slice/sssd.service
             ├─6043 /usr/sbin/sssd -i --logger=files
             └─6044 /usr/libexec/sssd/sssd_be --domain Sharkiplier.co --uid 0 --gid 0 --logger=files

Dec 20 13:35:30 Source.Sharkiplier.co systemd[1]: Starting System Security Services Daemon...
Dec 20 13:35:30 Source.Sharkiplier.co sssd[6043]: Starting up
Dec 20 13:35:30 Source.Sharkiplier.co sssd_be[6044]: Starting up
Dec 20 13:35:31 Source.Sharkiplier.co systemd[1]: Started System Security Services Daemon.
Dec 20 13:35:41 Source.Sharkiplier.co sssd[6062]: ; TSIG error with server: tsig verify failure
Dec 20 13:35:41 Source.Sharkiplier.co sssd[6062]: ; TSIG error with server: tsig verify failure
Dec 20 13:35:41 Source.Sharkiplier.co sssd[6066]: ; TSIG error with server: tsig verify failure
Dec 20 13:35:41 Source.Sharkiplier.co sssd[6066]: ; TSIG error with server: tsig verify failure
Dec 20 13:35:41 Source.Sharkiplier.co sssd[6070]: tkey query failed: GSSAPI error: Major = Unspecified >
lines 1-19/19 (END)
```

What is in yellow is what you are concerned with, and hopefully yours has a green little button as well.

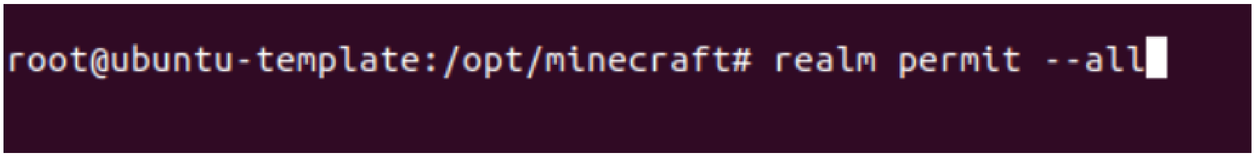
NOTE: as of now you almost done, however verify (below in the list) that it works because in this example I have the little green circle but I missed a download and it did not work.

Specifying who is allowed in

Based on how much you trust the AD security wise, you can allow different types of domain users in, such as only admins etc.

In this example I allow all because I trust that the AD is secure

☐ realm permit -- all

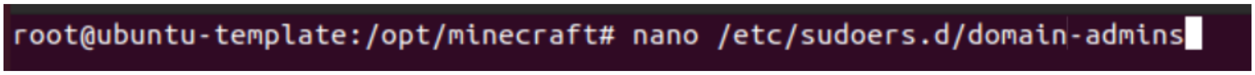


```
root@ubuntu-template:/opt/minecraft# realm permit --all
```

Specifying the permissions

If the domain admins need sudo privs on the box then you can give it to them by making a config file

□ nano /etc/sudoers.d/domain-admins



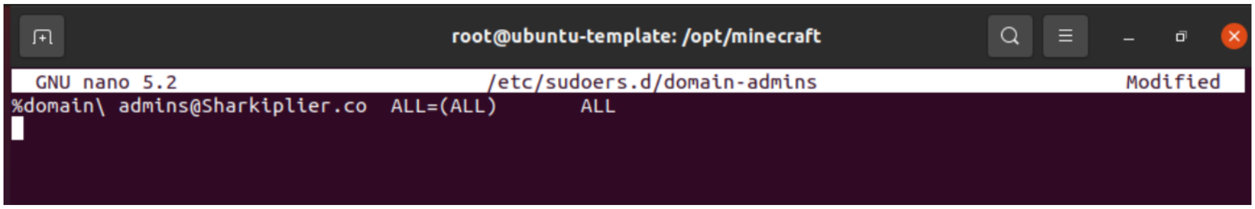
```
root@ubuntu-template:/opt/minecraft# nano /etc/sudoers.d/domain-admins
```

This file should be completely empty, as it is new and you just made it.

Type the following into the file and save it, for domain admins

□ %domain\ admins@{your-domain}   ALL=(ALL)   ALL

This command specifies what permissions they have, the ALL's here give them sudo privilages for any situation, which should be fine if they are domain admins anyway.



```
  GNU nano 5.2                     /etc/sudoers.d/domain-admins                        Modified
%domain\ admins@Sharkiplier.co  ALL=(ALL)       ALL
```

What the ending file should look like

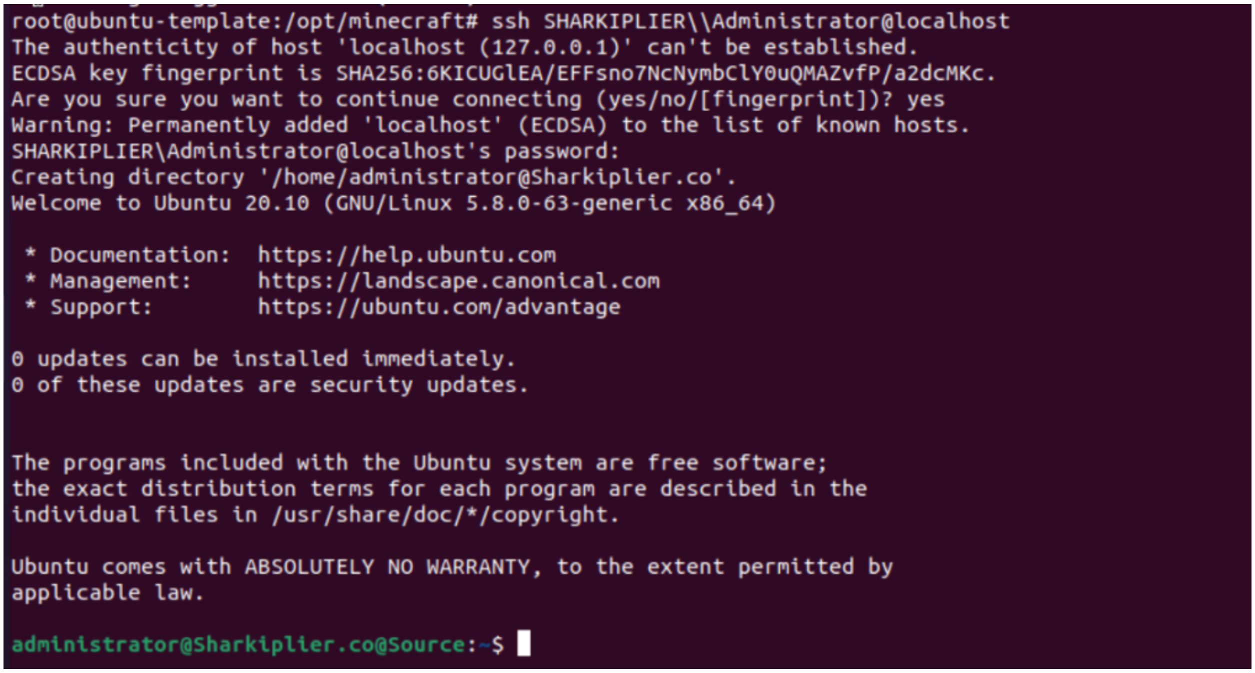
Verifying that you are able to connect

You are going to want to ssh into one of the accounts that are connected to the domain, in this case I am going to ssh into administrator

□ ssh {your-domain}\\{your-account}@localhost

this command ssh's into the domain with a specific command.

Normally this will work, you will sign in and you will see an image that looks like this

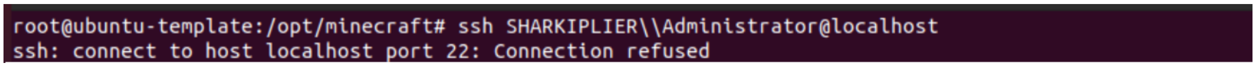


```
root@ubuntu-template:/opt/minecraft# ssh SHARKIPLIER\\Administrator@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:6KICUGlEA/EFFsno7NcNymbClY0uQMAZvfP/a2dcMKc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
SHARKIPLIER\Administrator@localhost's password:
Creating directory '/home/administrator@Sharkiplier.co'.
Welcome to Ubuntu 20.10 (GNU/Linux 5.8.0-63-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

administrator@Sharkiplier.co@Source:~$ █
```

And you are done!!

BugFixing (UNECESSARY IF YOURS WORKS)

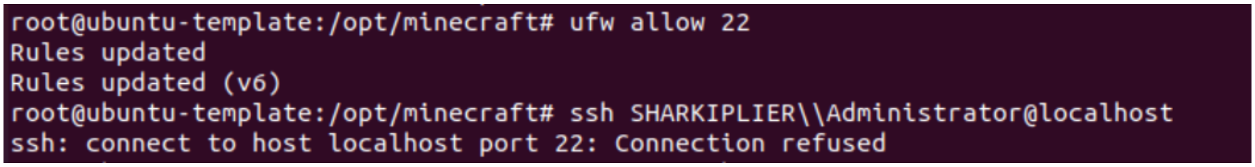
In my case I received an error that looks like this



```
root@ubuntu-template:/opt/minecraft# ssh SHARKIPLIER\\Administrator@localhost
ssh: connect to host localhost port 22: Connection refused
```

SSH is not working for whatever reason

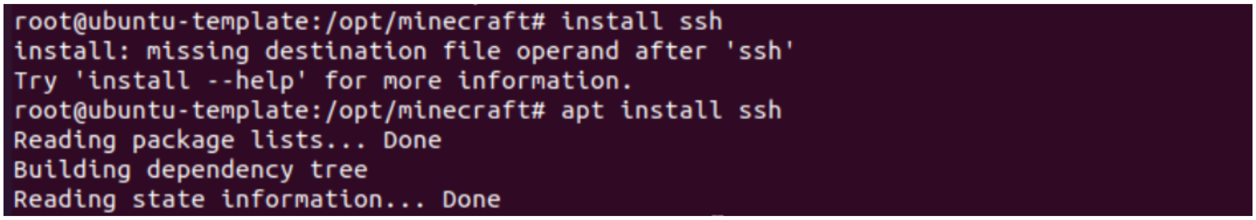
My first thought is that I blocked it with the firewall, so I allow it through UFW



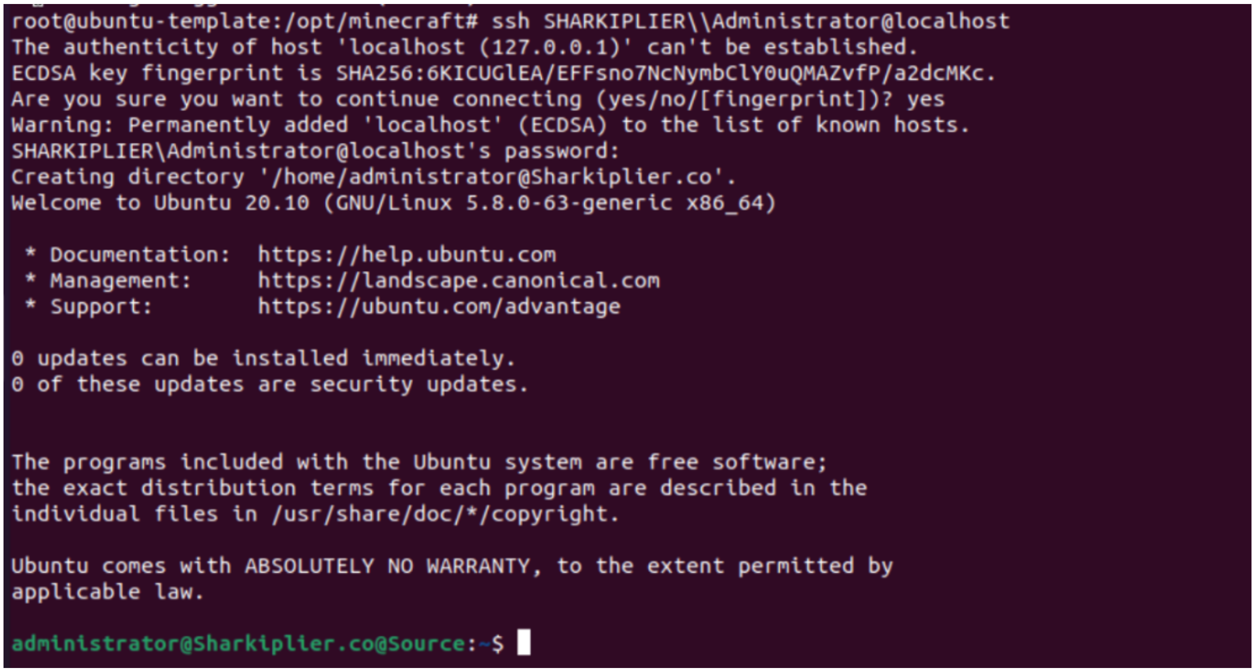
```
root@ubuntu-template:/opt/minecraft# ufw allow 22
Rules updated
Rules updated (v6)
root@ubuntu-template:/opt/minecraft# ssh SHARKIPLIER\\Administrator@localhost
ssh: connect to host localhost port 22: Connection refused
```

but as you can see, it still is not working, so I am thinking maybe there is an error in the packages of ssh

turns out I just did not install it, so here is me doing that and getting it working.



```
root@ubuntu-template:/opt/minecraft# install ssh
install: missing destination file operand after 'ssh'
Try 'install --help' for more information.
root@ubuntu-template:/opt/minecraft# apt install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
```



```
root@ubuntu-template:/opt/minecraft# ssh SHARKIPLIER\\Administrator@localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:6KICUGlEA/EFFsno7NcNymbClY0uQMAZvfP/a2dcMKc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
SHARKIPLIER\Administrator@localhost's password:
Creating directory '/home/administrator@Sharkiplier.co'.
Welcome to Ubuntu 20.10 (GNU/Linux 5.8.0-63-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

administrator@Sharkiplier.co@Source:~$
```