Setting up SSH for remote admin (RSA and Ed25519)
{Example from Linux Administration Cookbook}

RSA
Make sure you have a connection to your VM
$ ping 192.168.33.11

Generate a key pair on your host machine. The public key will be copied onto your server and private key will be on your local machine.
$ ssh-keygen -b 4096 -C “Example RSA Key”
Copy your public key onto your VM. Once your public key is on your server, you can ssh into your server using your private key on your local machine.
$ ssh-copy-id 192.168.33.11

```
[vagrant@centos1 ~]$ ssh-keygen -b 4096 -C "Example RSA Key"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vagrant/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vagrant/.ssh/id_rsa.
Your public key has been saved in /home/vagrant/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:uTvbKVe2XRHyK9DuQH10DzB23CC+6MG/bkhOyn9jXIA Example RSA Key
The key's randomart image is:
+---[RSA 4096]----+
|            =ooo |
|           o =ooo|
|           .+ +.+|
|         oE+.+ +.|
|        S = +.. o|
|         oo+oo...|
|       ..=.=++.. |
|        =o+.Bo.  |
|        o*+=o.   |
+----[SHA256]-----+
[vagrant@centos1 ~]$ ssh-copy-id 192.168.33.11
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/vagrant/.ssh/id_rsa.pub"
The authenticity of host '192.168.33.11 (192.168.33.11)' can't be established.
ECDSA key fingerprint is SHA256:muueS4aYL18TM20OtmCUhsovqfOWWk4wLtmIPCYO9b4.
ECDSA key fingerprint is MD5:df:6a:d1:d1:30:79:db:1f:48:c4:4e:3a:21:62:7b:23.
Are you sure you want to continue connecting (yes/no)? 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
vagrant@192.168.33.11's password:
Permission denied, please try again.
vagrant@192.168.33.11's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh '192.168.33.11'"
and check to make sure that only the key(s) you wanted were added.
```

Check to see if you can ssh into your VM via key
$ ssh 192.168.33.11

```
[vagrant@centos1 ~]$ ssh 192.168.33.11
[vagrant@centos2 ~]$
```

Ed25519

Make sure you have a connection to your VM

$ ping 192.168.33.11

Generate Ed25519 key

$ ssh-keygen -t ed25519 -C “Example Ed25519 Key”

```
[vagrant@centos1 ~]$ ssh-keygen -t ed25519 -C "Example Ed25519 Key"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/vagrant/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vagrant/.ssh/id_ed25519.
Your public key has been saved in /home/vagrant/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:6t984CFsyRqoAZHHFwNYfcJpQGr1i1G4AGJw3Zkzp2M Example Ed25519 Key
The key's randomart image is:
+--[ED25519 256]--+
|*B*B+= o         |
|*++.@.B .        |
|.+.+.+ =         |
|o  .o E          |
| . . + +S.       |
|  . . ..* o      |
|   o  .+ o o     |
|  .  ..  o. .    |
|      ... o.     |
+----[SHA256]-----+
```

Copy your newly created public key to your VM (192.168.33.11)

$ ssh-copy-id -i .ssh/id_ed25519.pub 192.168.33.11

```
[vagrant@centos1 ~]$ ssh-copy-id -i .ssh/id_ed25519.pub 192.168.33.11
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/id_ed255
19.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to fil
ter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are pr
ompted now it is to install the new keys

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh '192.168.33.11'"
and check to make sure that only the key(s) you wanted were added.
```

Check to see if you can ssh into your VM using your new Ed25519 key

```
[vagrant@centos1 ~]$ ssh 192.168.33.11 -i .ssh/id_ed25519
Last login: Tue Mar 24 23:48:55 2020 from 192.168.33.10
```

****Make sure your ssh permissions are correct:**
Public keys (with .pub) have 644 perms
Private keys have 600 perms