Creating a Linux VM

(Because Linux is better, that’s why)

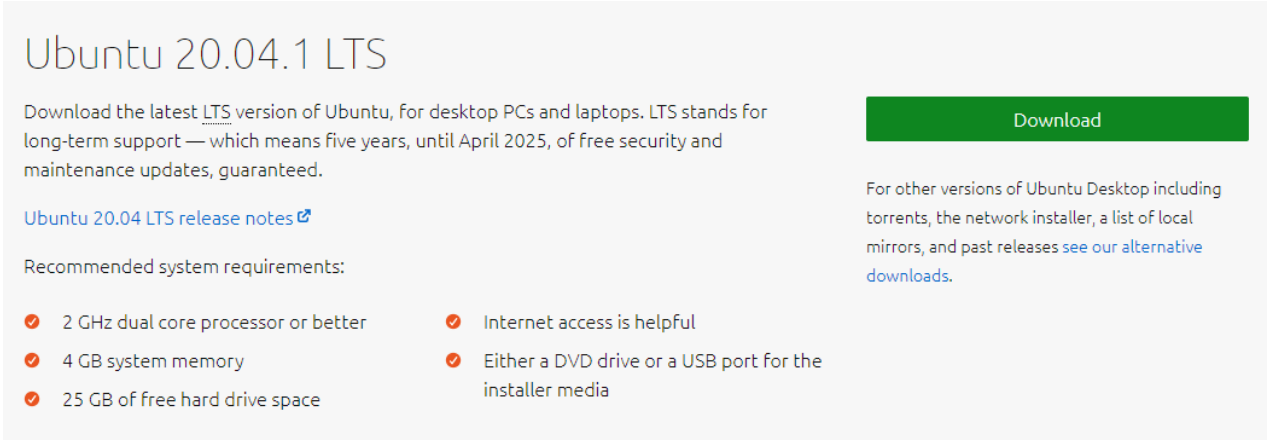
Overview

1. Choose and download your distribution software
2. Create your VM using VMware Workstation
3. Launch your VM and perform initial startup
4. Install Ignition and Designer Launcher
5. Install SQL Server and MSSQL-Tools

Step 1. Choose and download your distribution

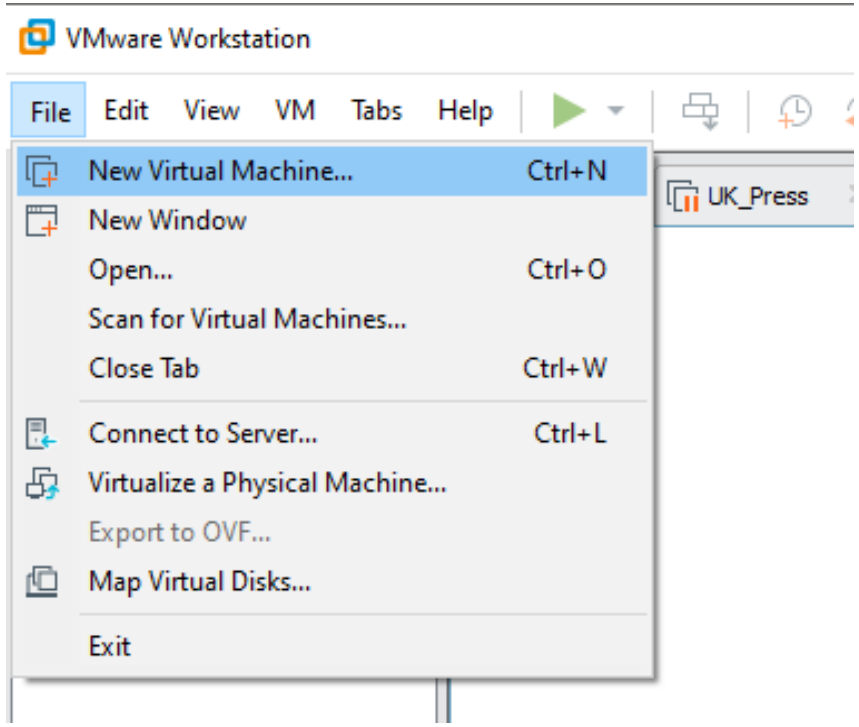
- I like Ubuntu. https://ubuntu.com/download
- Debian is also a solid choice. https://www.debian.org/distrib/netinst
- There are thousands of others. Mint and RedHat are also popular, but you'll have the most support online for specific problems with Ubuntu or Debian.
- Unless you're super comfortable with Linux, I recommend sticking to a popular one.
- For this guide, I will be using Ubuntu 20.04.1 LTS.

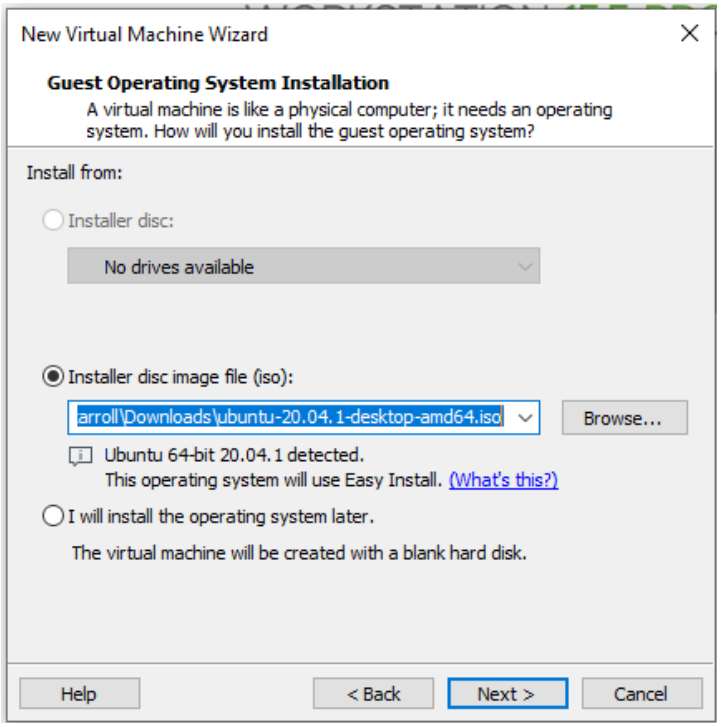
Look at how ridiculously low these requirements are! →



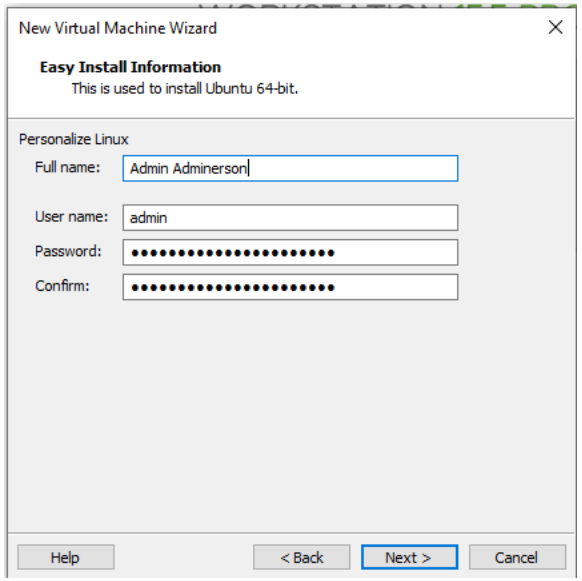
Step 2. Create your VM using VMware
(or any other software you choose)

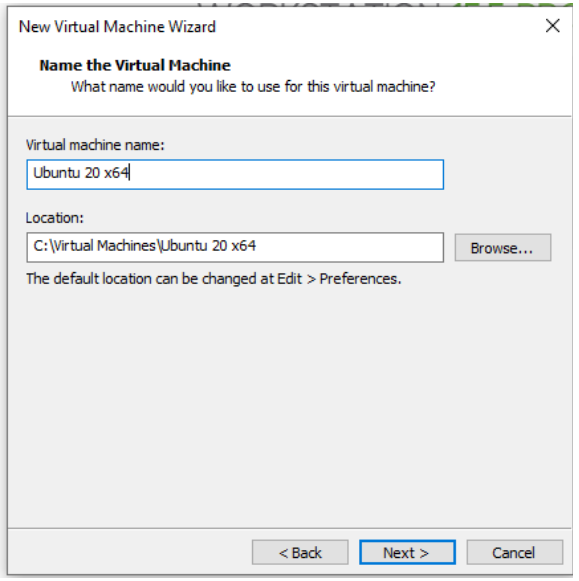
- Create a blank machine and follow wizard to set up OS, Memory, CPU, etc.

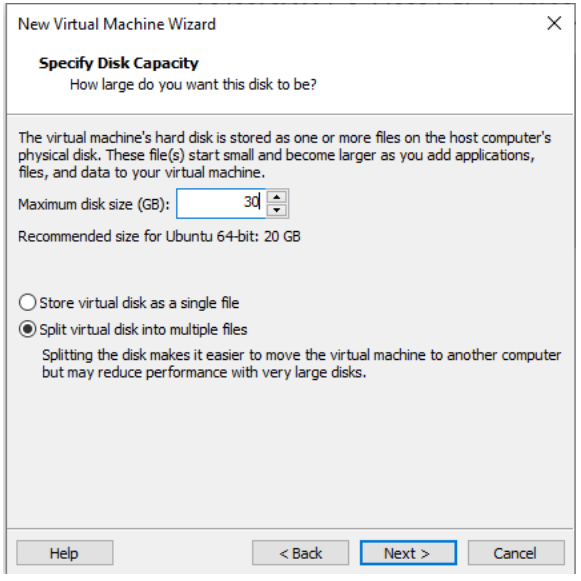


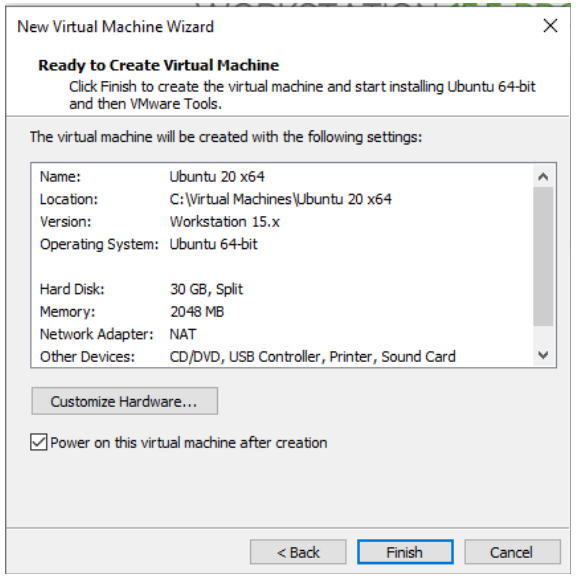


Step 2. Create your VM using VMware



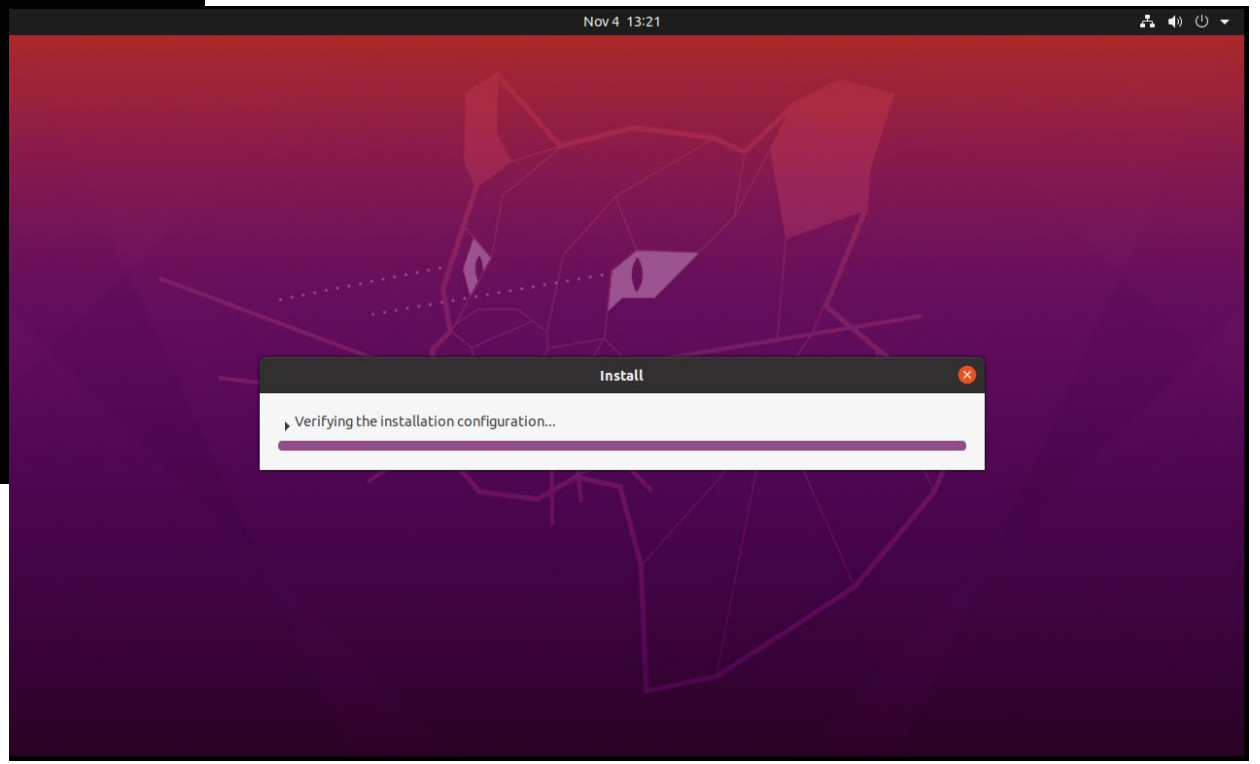




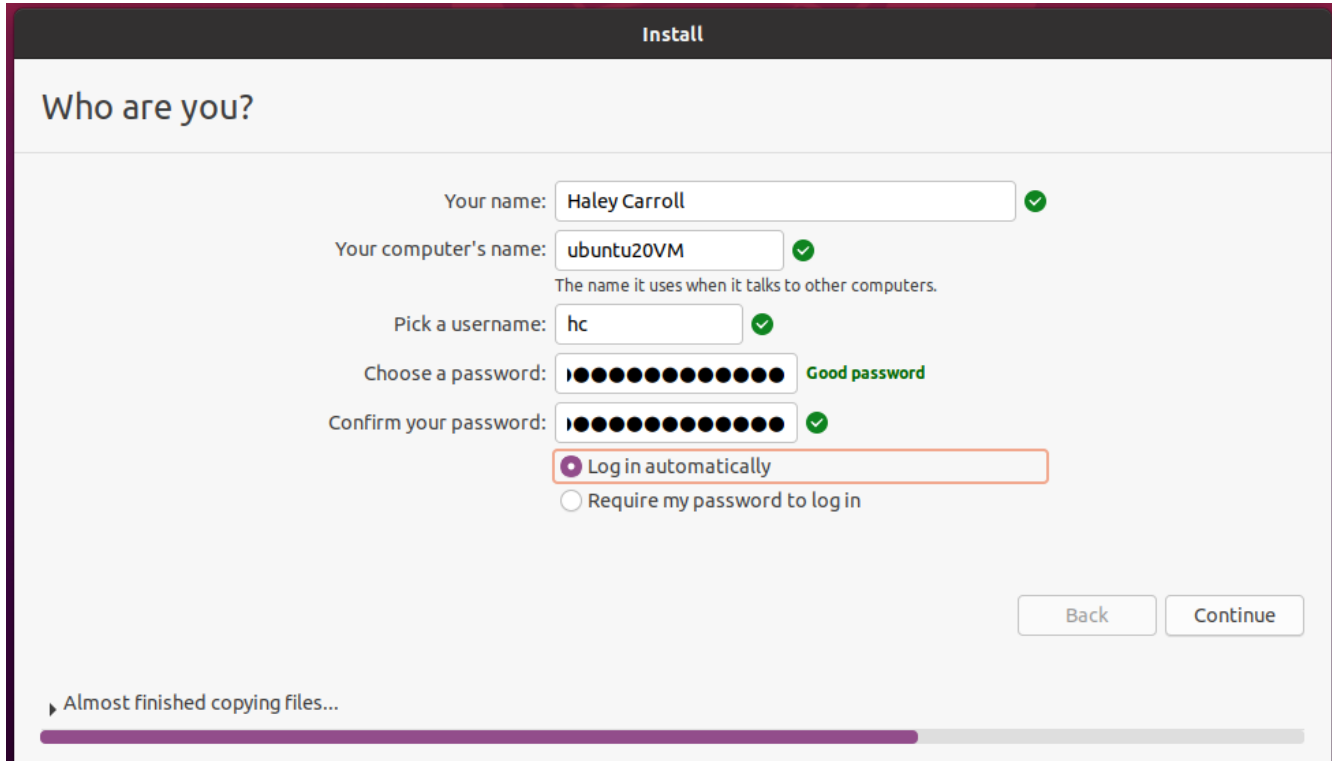


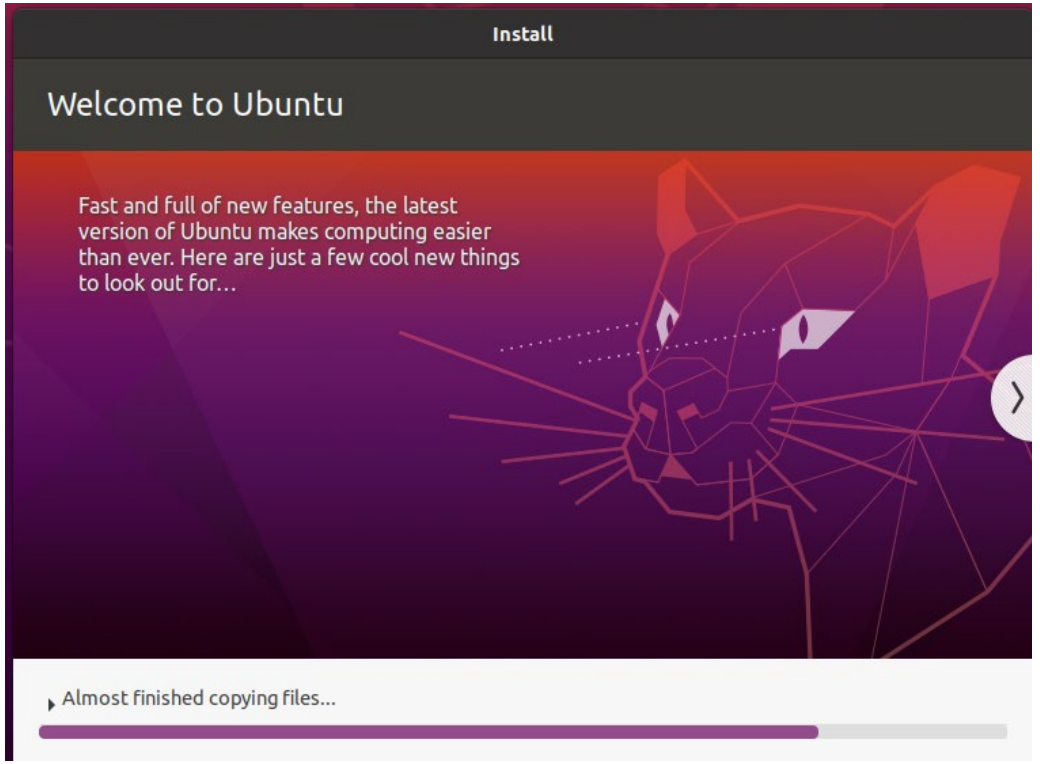
- After creation, increase the memory of the VM for better performance.
- You can also increase the number of processors or add more cores.
- This can only be done while the VM is shut down.

Step 3. Launch VM and perform initial startup



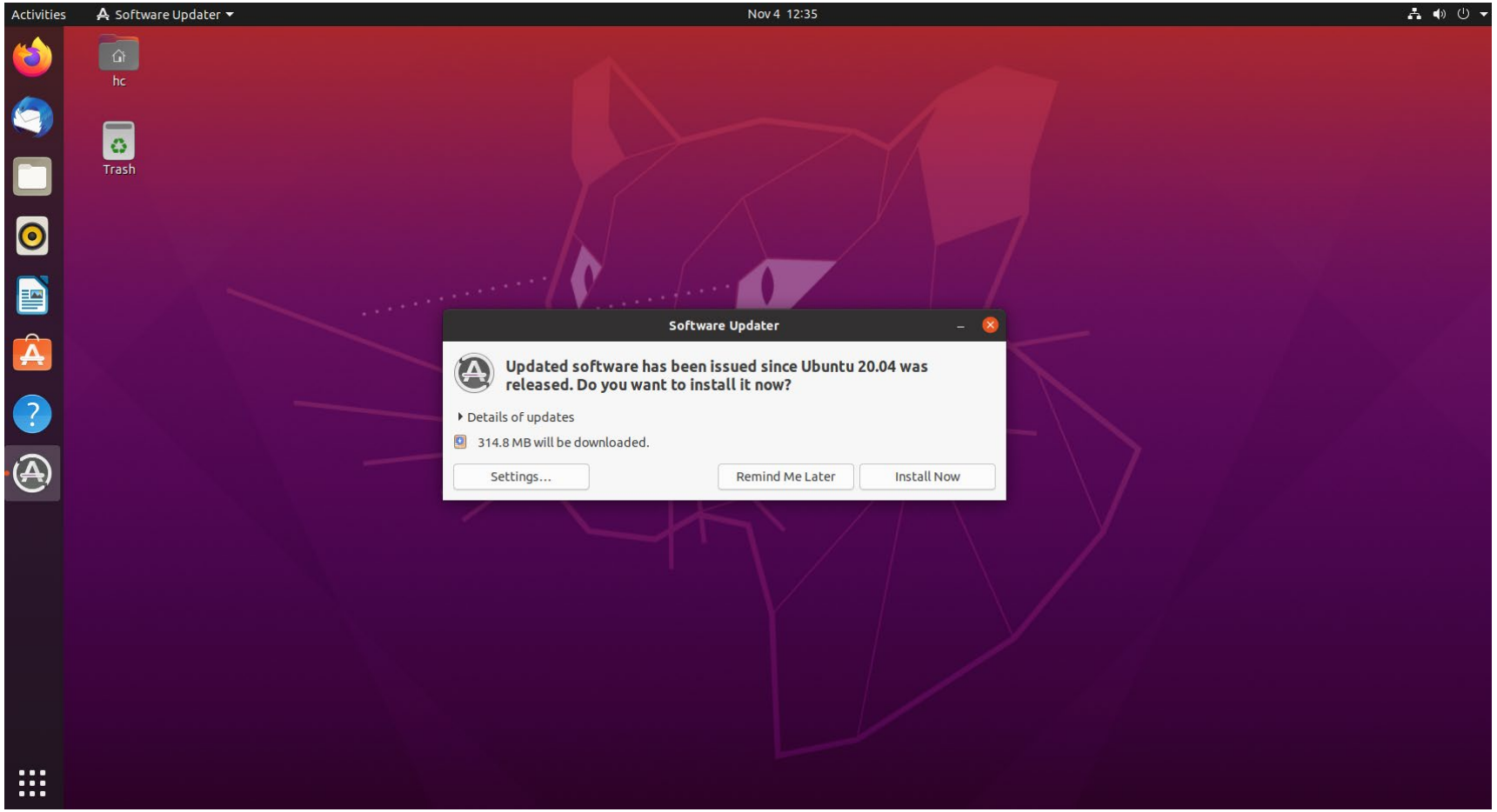
Step 3. Launch VM and perform initial startup





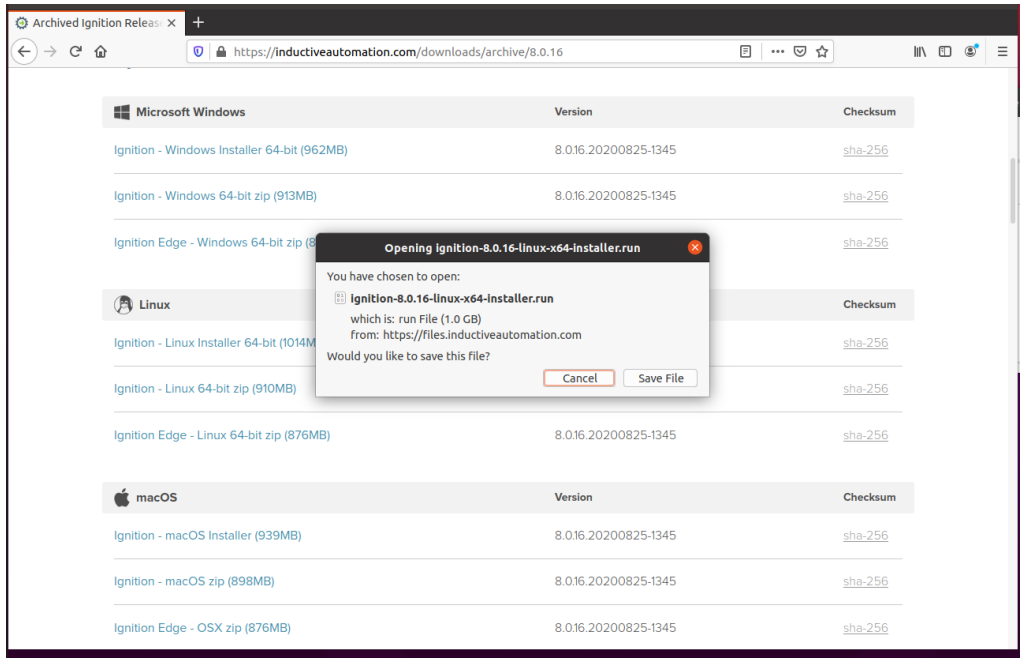
So far, the longest part of this process was downloading the iso…

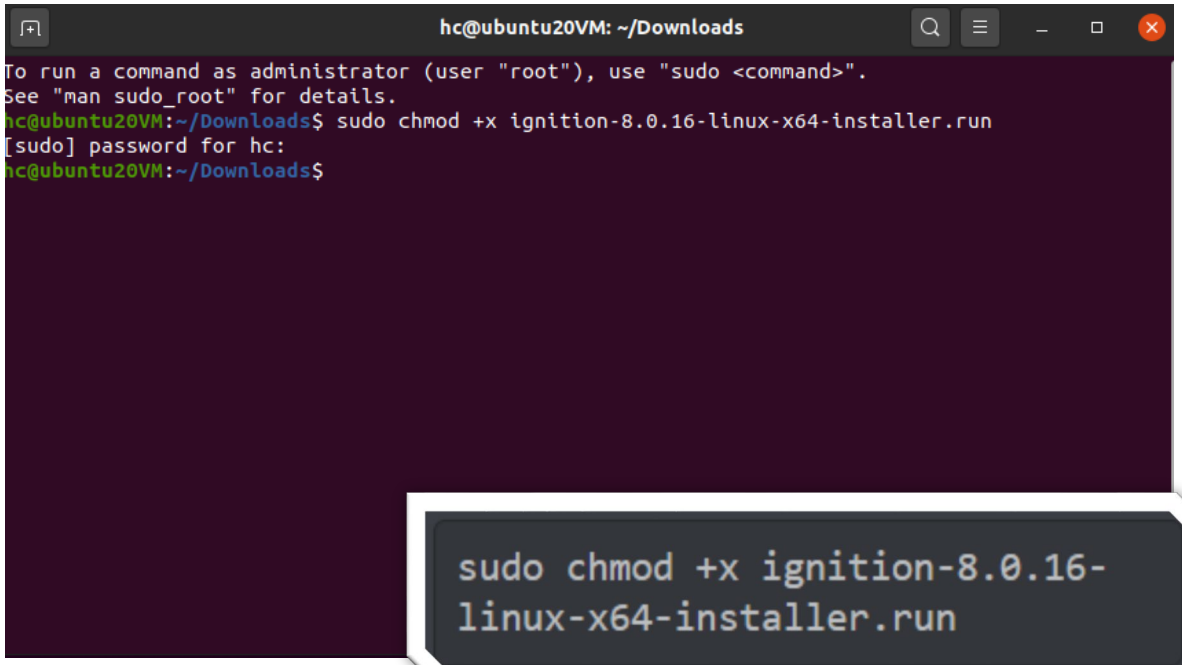
Step 3. Launch VM and perform initial startup



Step 4. Install Ignition

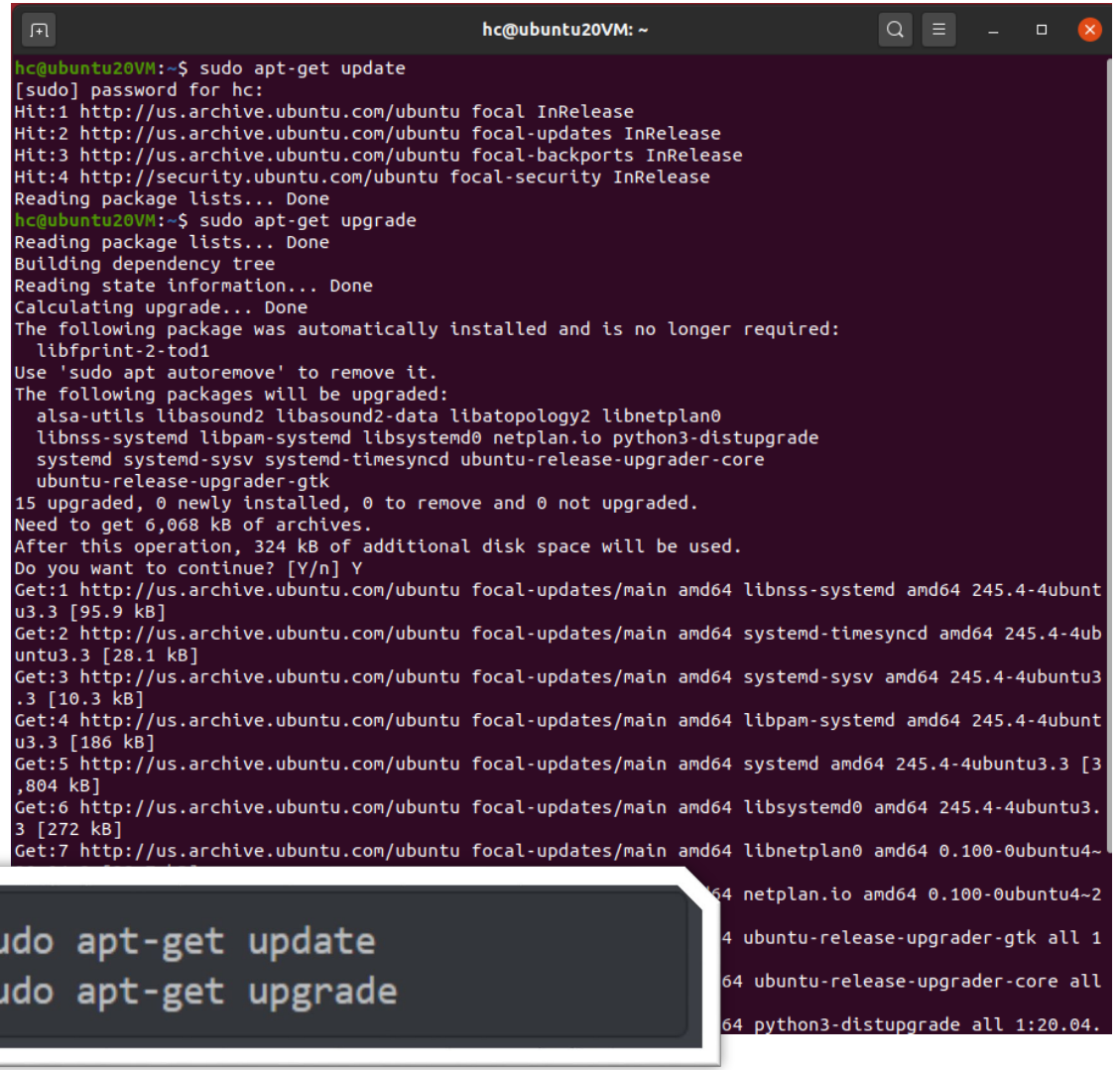
- We'll go through installing Ignition 8.
- Once downloaded use a chmod command to make the installer useable.
- Ignition works great in Linux. If you're lucky, you might also encounter a rare Linux customer.





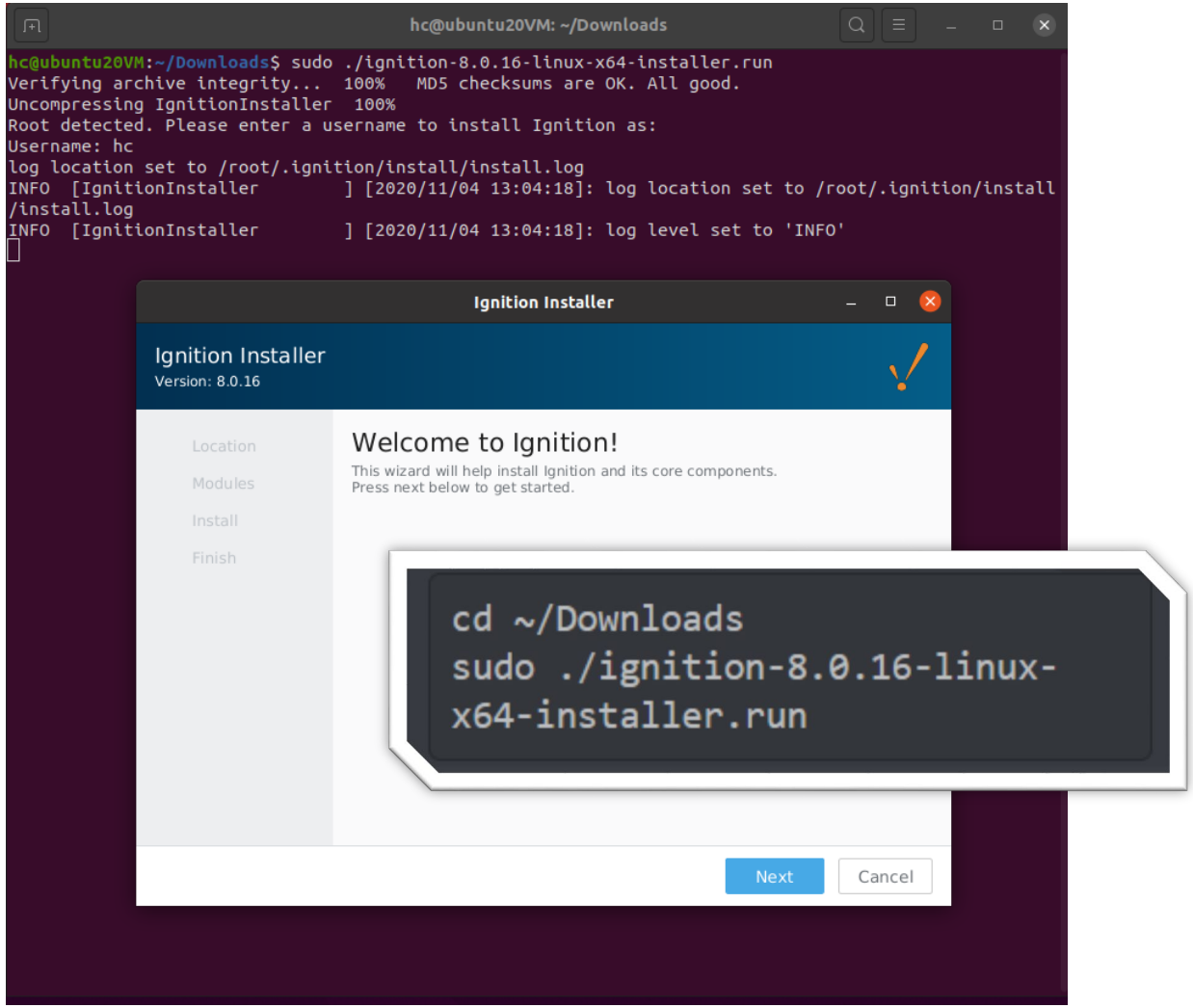
Step 4. Install Ignition

- Update the system to make sure everything is installed.



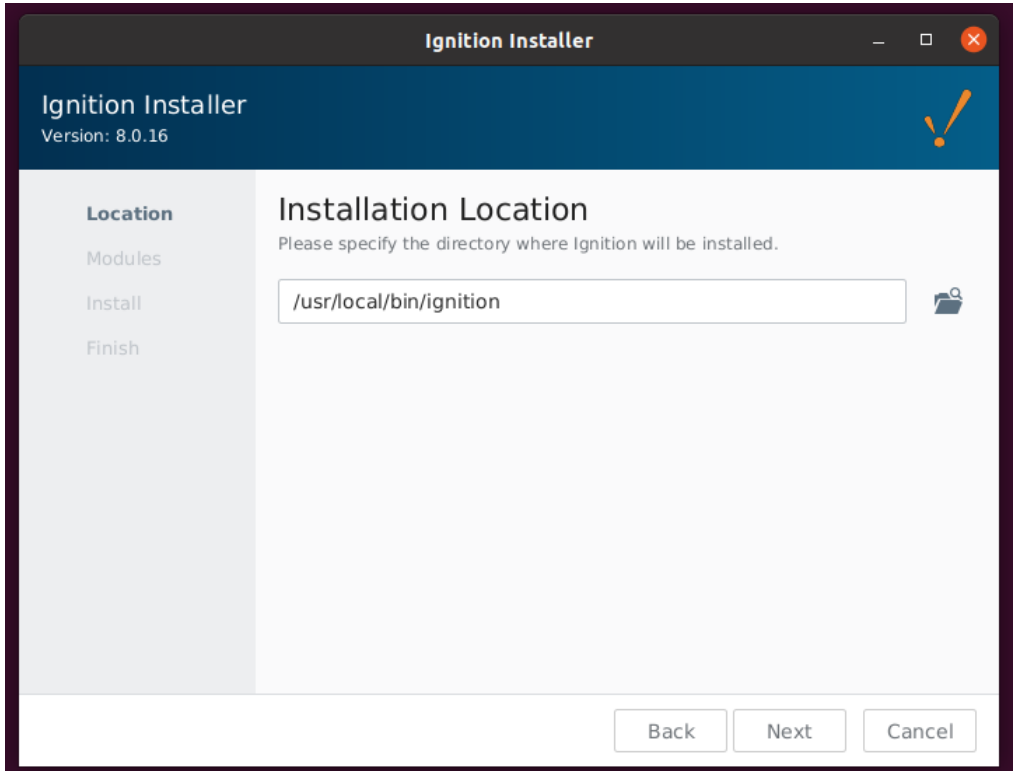
Step 4. Install Ignition

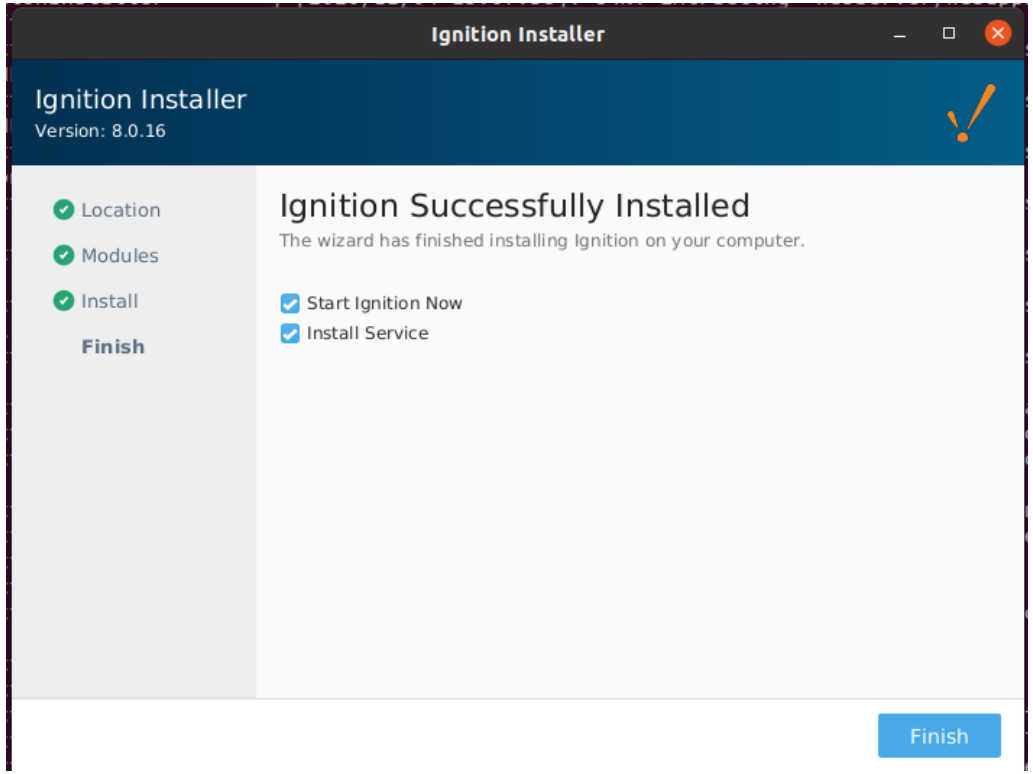
- Switch to the directory where you downloaded the installer and run it.



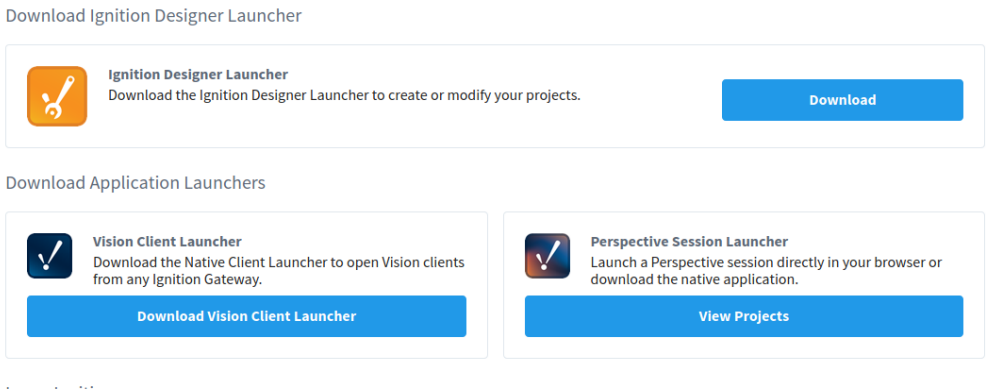
Step 4. Install Ignition

- Since we didn't specify to run it in text mode, the familiar UI installation will pop up.
- The default installation location for Linux is /usr/local/bin/Ignition. This is analogous to C:\Program Files\Inductive Automation\Ignition

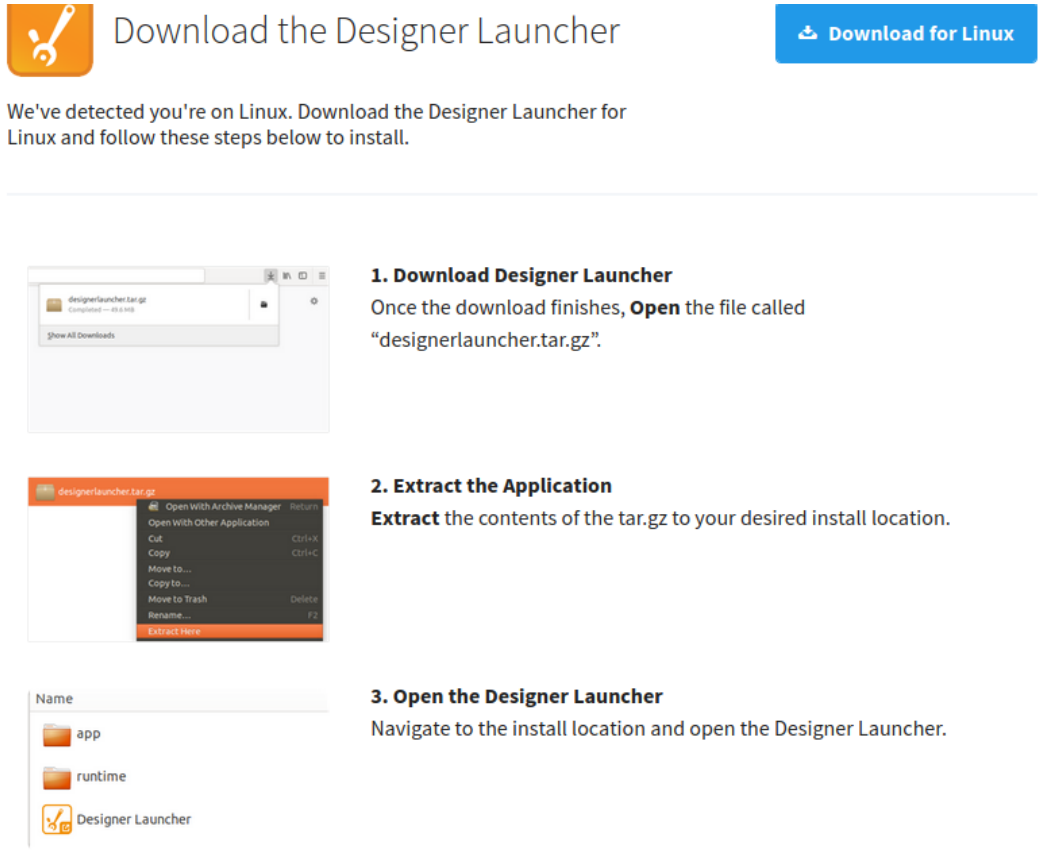




Step 4.5. Download the Designer Launcher

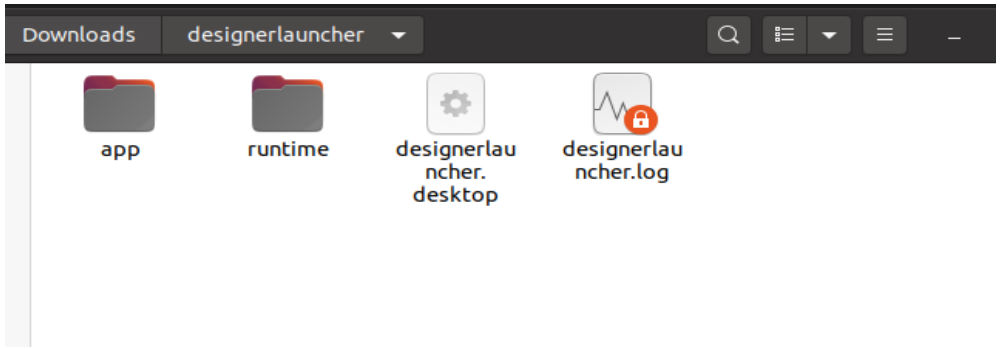


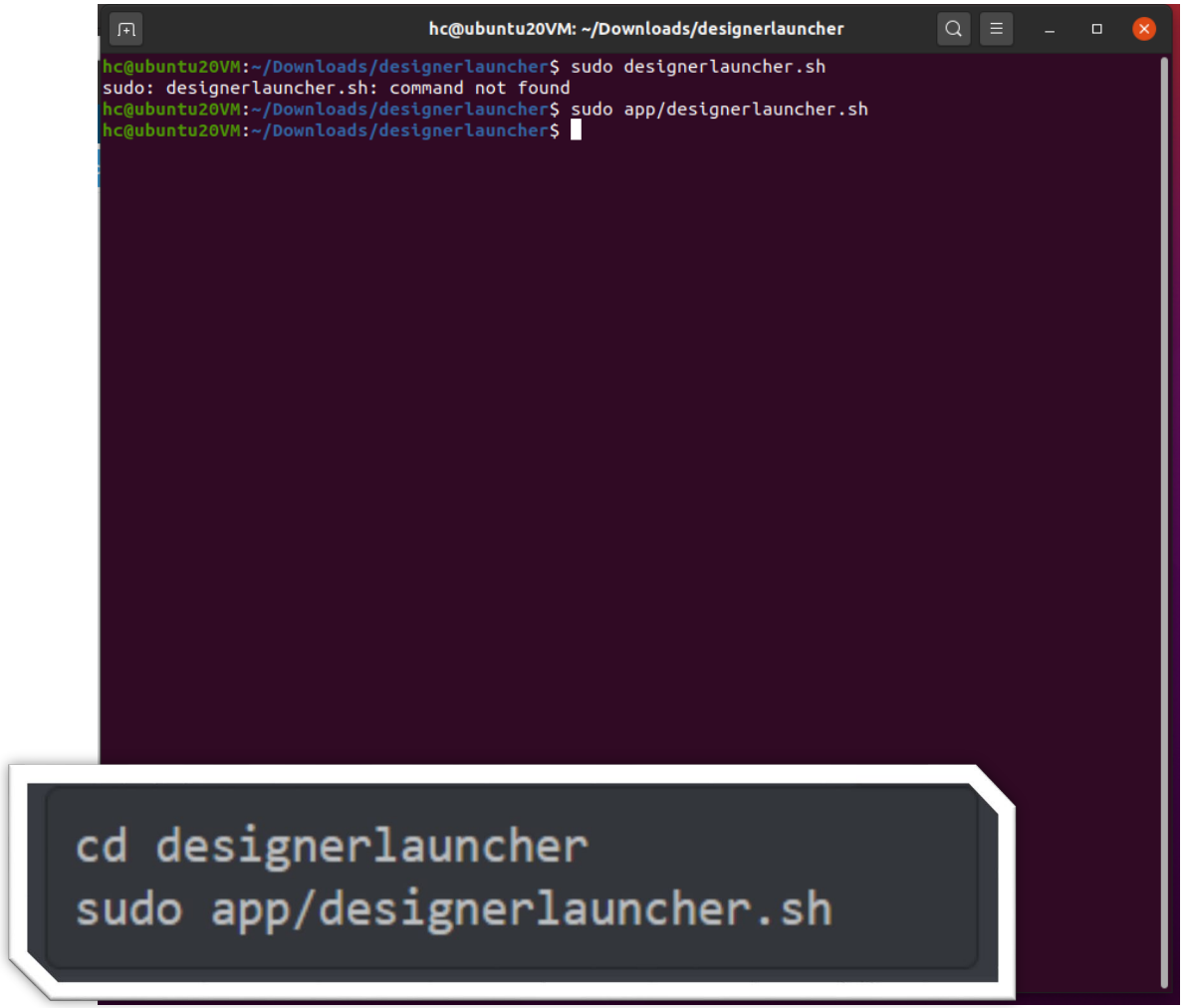
These instructions aren't totally accurate, so we'll go through that here.



Step 4.75. Make Designer Launcher work

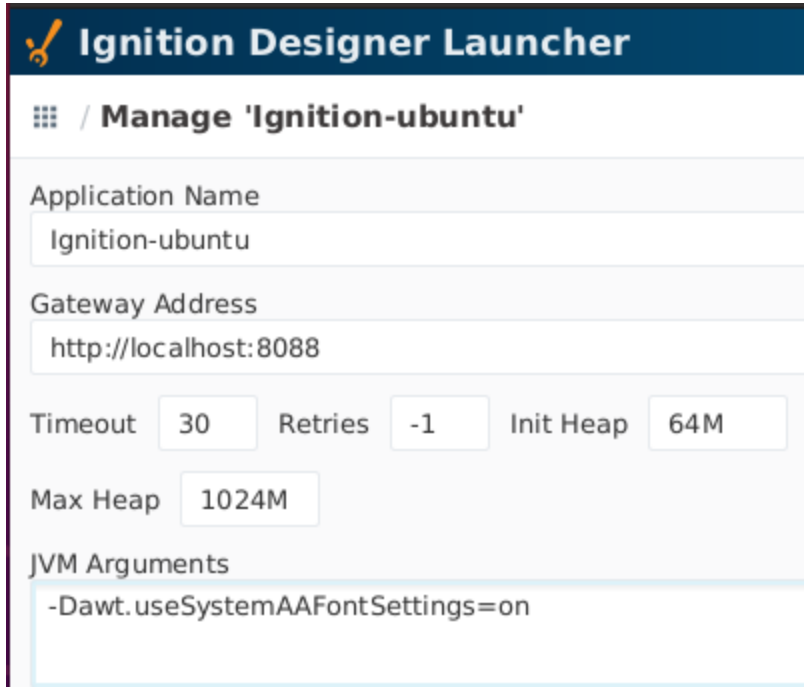
- Extract the tar.gz file to wherever you want.
- Go to that directory and enter the extracted designerlauncher folder.
- Run sudo app/designerlauncher.sh which will launch the Designer Launcher.
- If you want to have the Designer Launcher executable from a file, you will need to write a script that will do these commands.

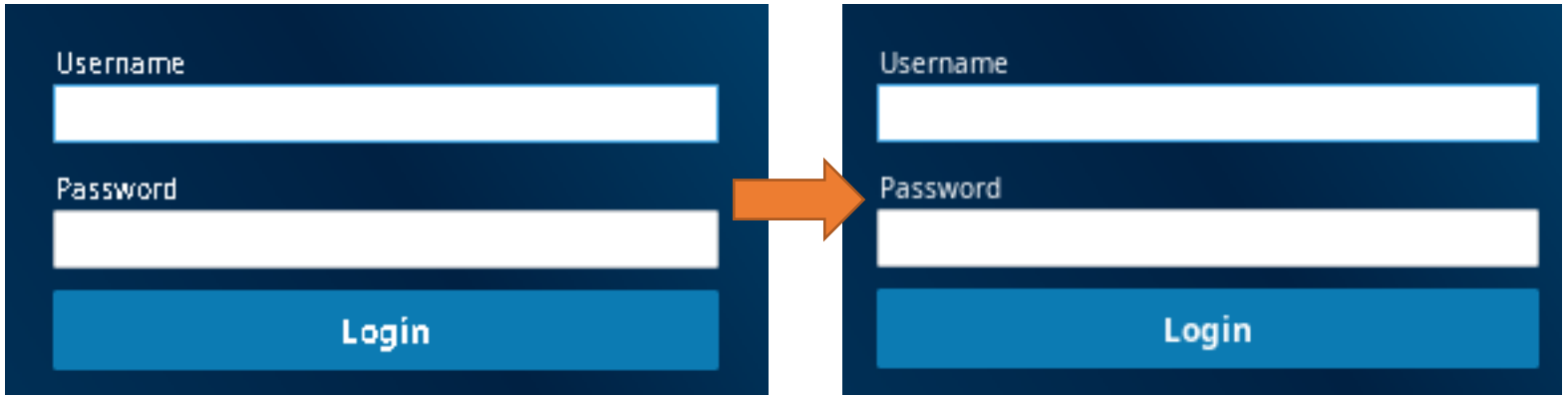




Step 4.99. Force nicer fonts

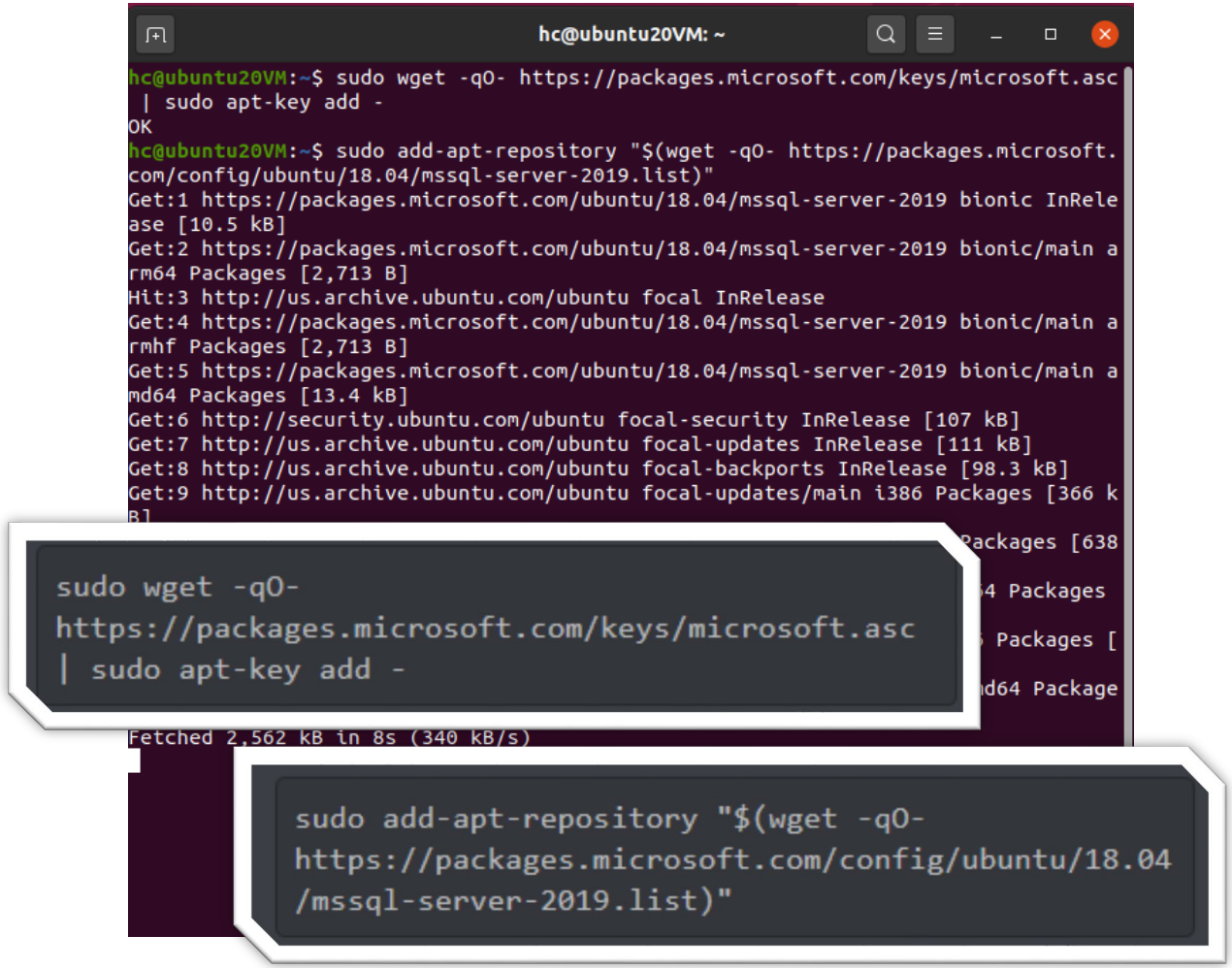
- For some reason, Java doesn't set fonts to be anti-aliased by default.
- In the Designer Launcher, click the 3 buttons on the right of the designer and select Manage.
- In the JVM Arguments add –Dawt.useSystemAAFontSettings=on





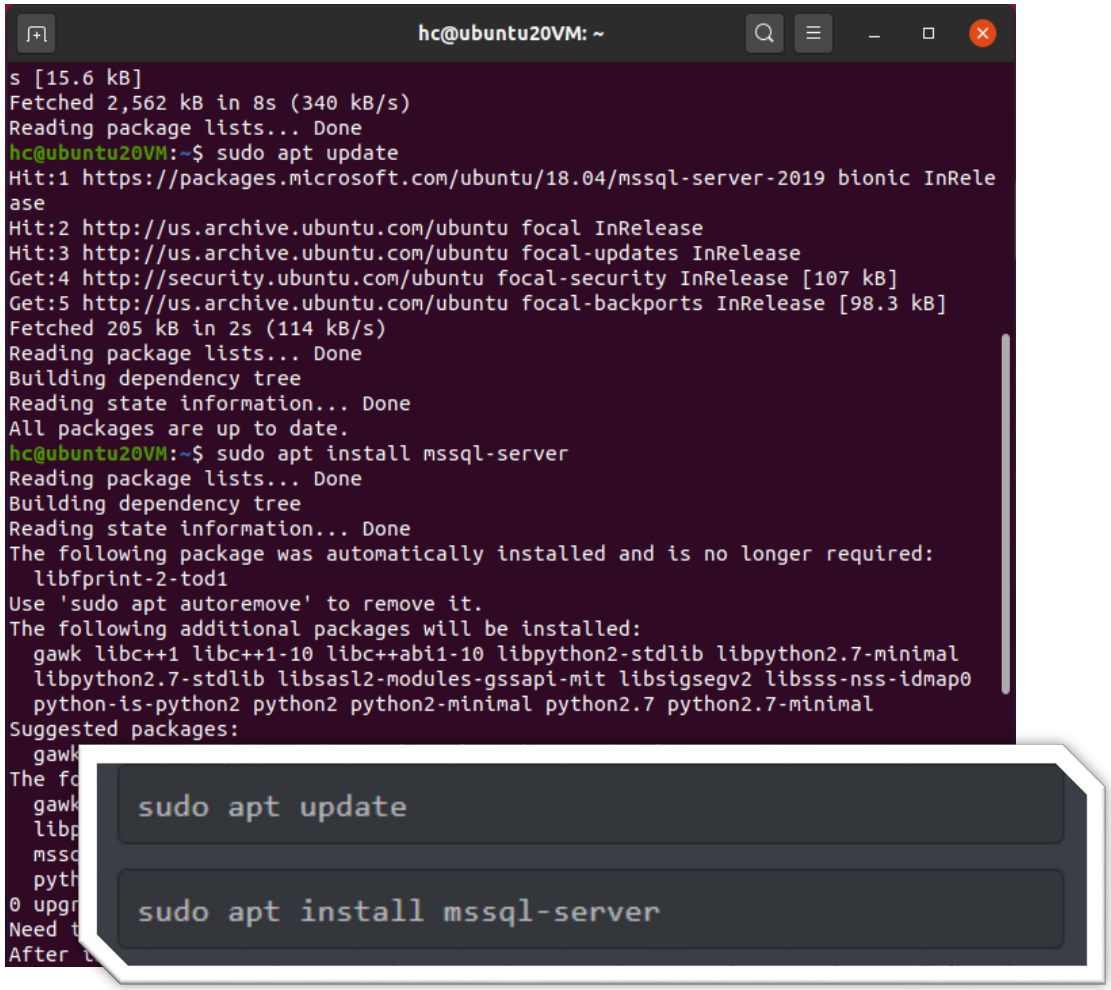
Step 5. Install SQL Server

- First, get the public keys from Microsoft.
- Then get the SQL server repository.



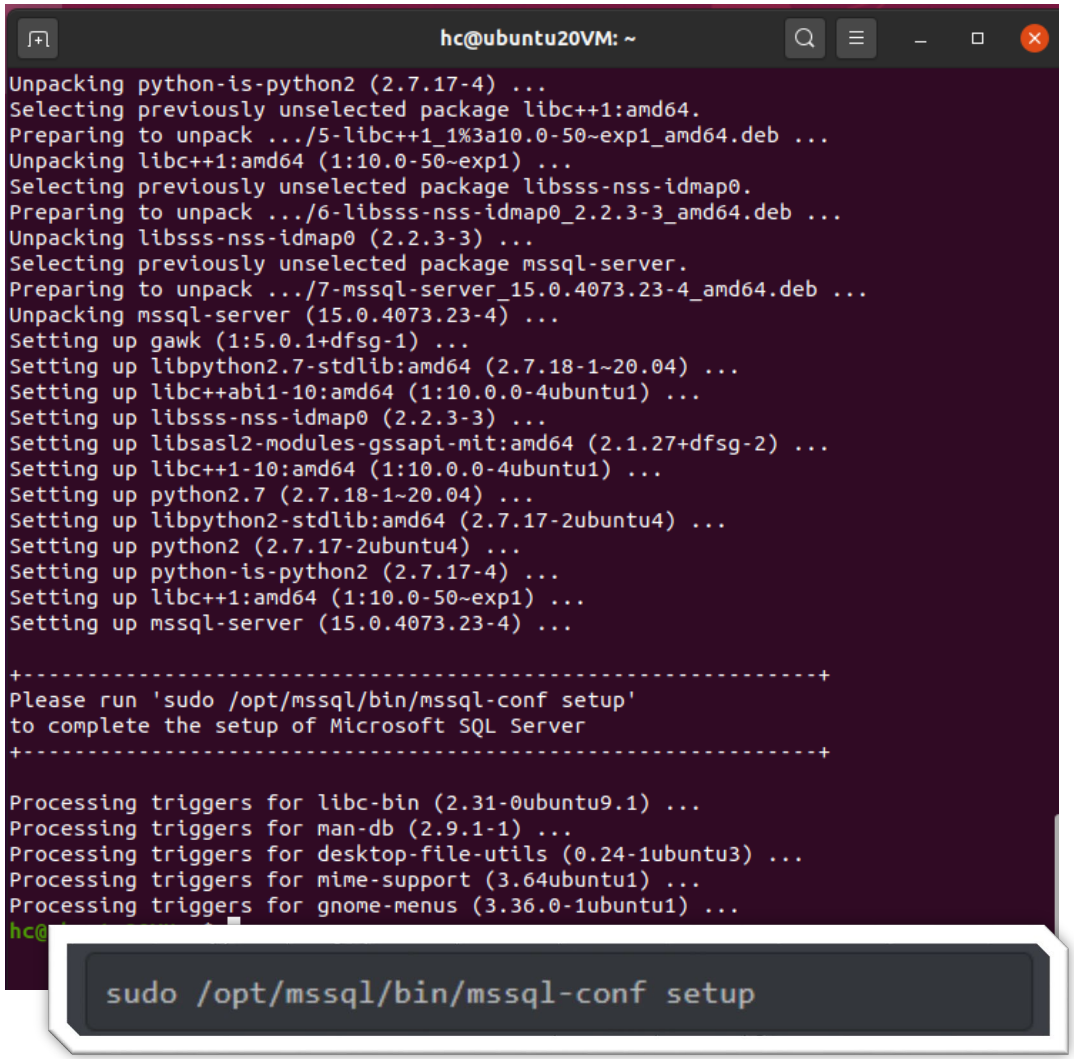
Step 5. Install SQL Server

- Update the system and install mssql-server.



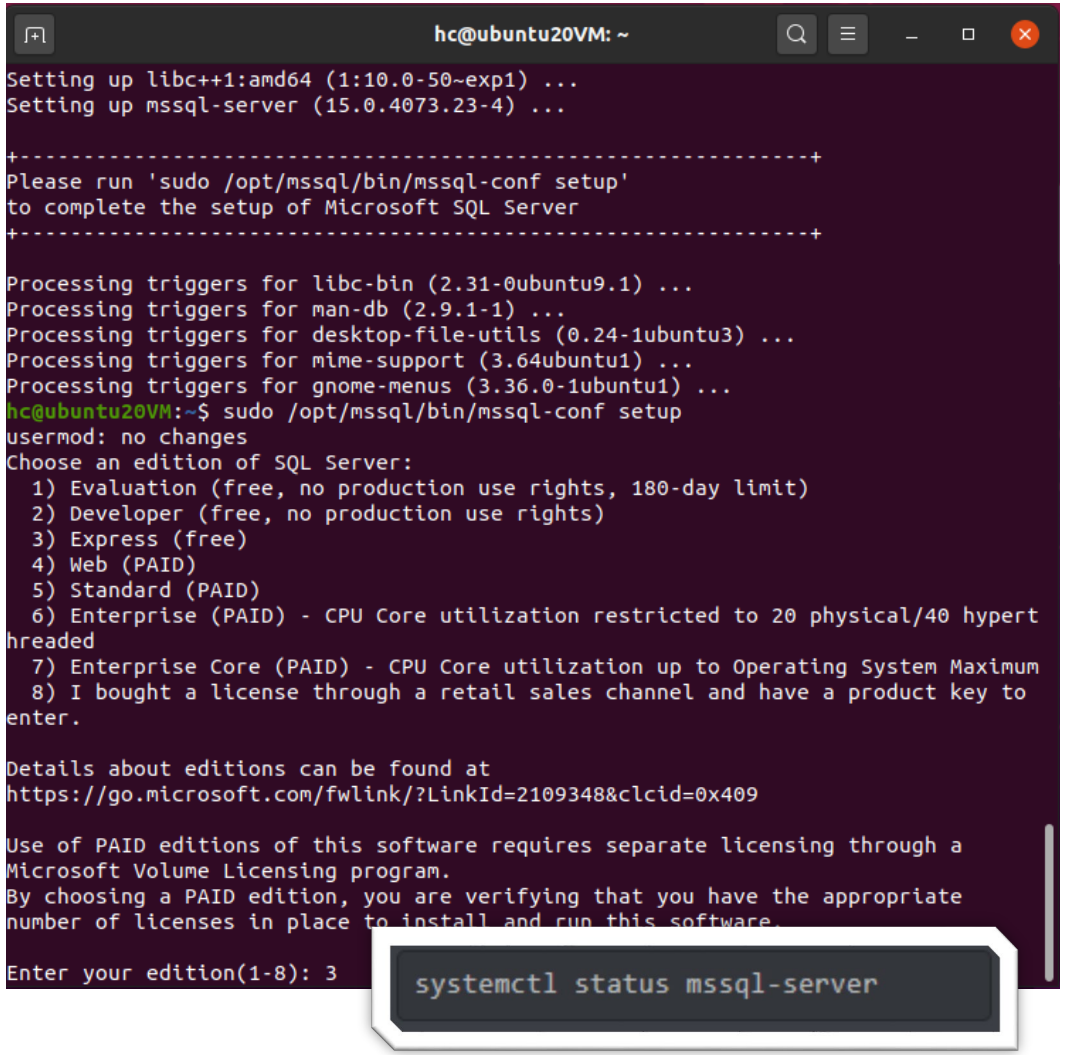
Step 5. Install SQL server

- Choose which version of SQL server to use and set up your SA password.



Step 5. Install SQL server

- Use the systemctl status command to check the status of SQL server and confirm that it is running.
- At this point, SQL is installed on the host machine and you should be able to connect to this database through SSMS.



Step 5.5. Install mssql-tools

```
hc@ubuntu20VM:~$ curl https://packages.microsoft.com/keys/microsoft.asc | sudo a
pt-key add -
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   983  100   983    0     0   2700      0 --:--:-- --:--:-- --:--:--  2700
OK
hc@ubuntu20VM:~$ curl https://packages.microsoft.com/config/ubuntu/19.10/prod.li
st > /etc/apt/sources.list.d/mssql-release.list
bash: /etc/apt/sources.list.d/mssql-release.list: Permission denied
```

```
sudo apt install curl
```

```
curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
```

- Mssql-tools allows sqlcmd, so you don't have to connect your host machine to ssms to set up.
- Curl is used to transfer data similar to FTP. Get the public keys from Microsoft and download mssql-tools.
- After that, we'll run sudo apt update, upgrade and then install mssql-tools.

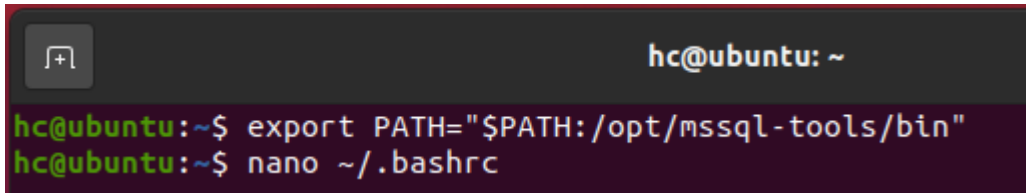
```
hc@ubuntu20VM:~$ curl https://packages.microsoft.com/config/ubuntu/19.10/prod.li
st | sudo tee /etc/apt/sources.list.d/msprod.list
[sudo] password for hc:   % Total    % Received % Xferd  Average Speed   Time
 Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    87  100    87    0     0    154      0 --:--:-- --:--:-- --:--:--   154

deb [arch=amd64,arm64,armhf] https://packages.microsoft.com/ubuntu/19.10/prod eo
an mainhc@ubuntu20VM:~$ sudo apt update
Hit:1 https://packages.microsoft.com/ubuntu/20.04/prod focal InRelease
Get:2 https://packages.microsoft.com/ubuntu/19.10/prod eoan InRelease [10.4 kB]
Hit:3 http://archive.ubuntu.com/ubuntu focal InRelease
Get:4 https://packages.microsoft.com/ubuntu/19.10/prod eoan/main amd64 Packages
[39.3 kB]
Get:5 https://packages.microsoft.com/ubuntu/19.10/prod eoan/main arm64 Packages
[844 B]
Get:6 https://packages.microsoft.com/ubuntu/19.10/prod eoan/main armhf Packages
[554 B]
Fetched 51.1 kB in 1s (67.6 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
hc@ubuntu20VM:~$ sudo apt install mssql-tool
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package mssql-tool
hc@ubuntu20VM:~$ sudo apt install mssql-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be inst
  libodbc1 msodbcsql17 odbcinst odbcinst1debia
Suggested packages:
  unixodbc-bin
The following NEW packages will be installed:
  libodbc1 msodbcsql17 mssql-tools odbcinst odbcins
0 upgraded, 6 newly installed, 0 to remove and 0 not
Need to get 1,632 kB of archives.
After this operation, 1,059 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

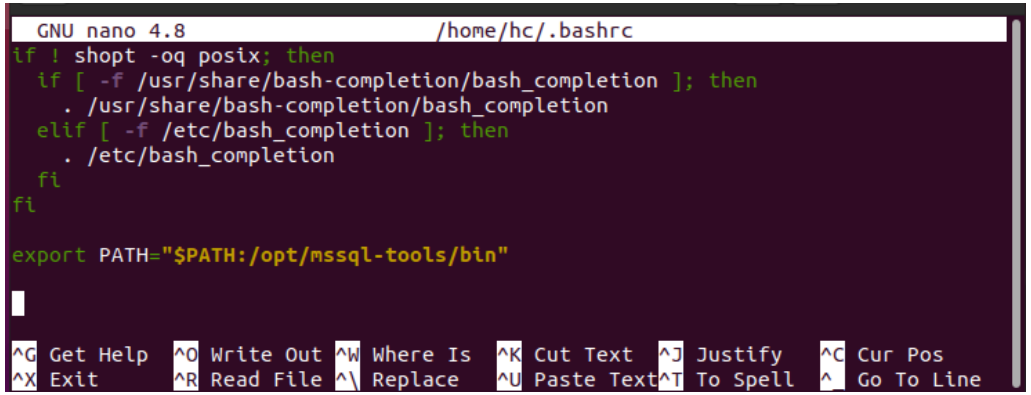
```
curl https://packages.microsoft.com/config/ubuntu/19.10/prod.list | sudo tee /etc/apt/sources.list.d/msprod.list
```

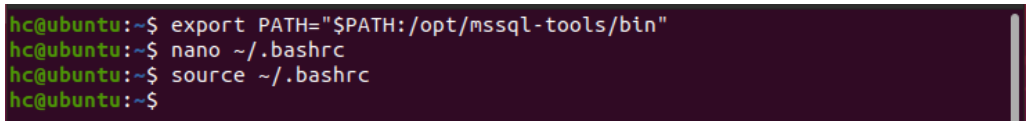
```
sudo apt update
```

```
sudo apt install mssql-tools
```

Step 5.75. Edit path for sqlcmd





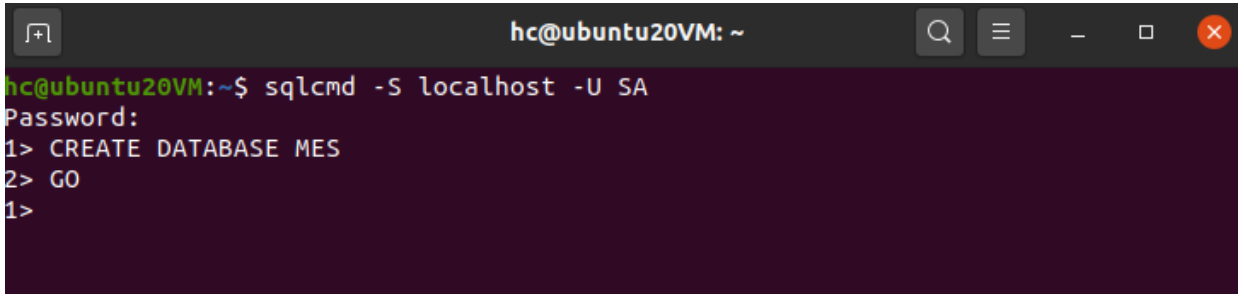


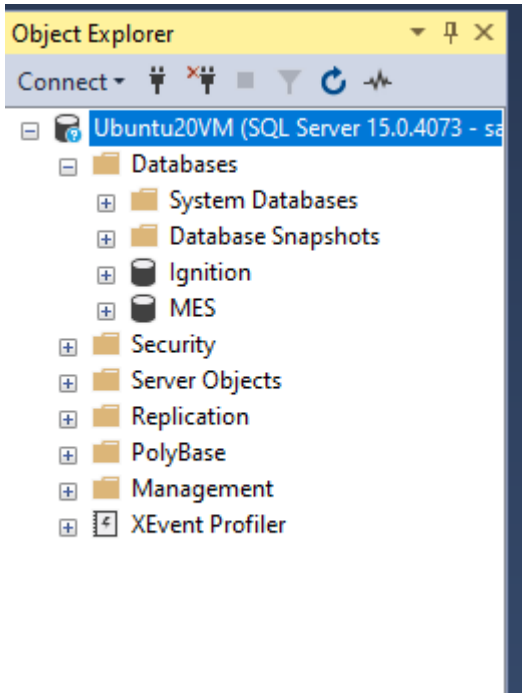

```
export PATH="$PATH:/opt/mssql-tools/bin"
nano ~/.bashrc
source ~/.bashrc
```

- Now, mssql-tools is installed. But unless we want to go to the installation directory every time, we'll need to add it to the path so it's accessible universally.
- To do that, enter the command export PATH="$PATH:/opt/mssql-tools/bin". That will make this terminal session run sqlcmd from any directory.
- Since we this to be automatic, edit the bashrc file to include that same statement. Bashrc is a script that runs whenever a terminal session is opened.
- Run source ~/.bashrc to execute it.

Step 5.75. Edit path for sqlcmd

- Now we can use sqlcmd to connect to our local instance.
- Create a new database and then verify it in SSMS on the host machine.
- We can also add this database connection to Ignition. It should come back as a valid connection.





Step 6. Laugh at people downloading 120 GB VMs

- Before you go back to work, make a copy of this VM.
- It’s only a few gigs and you’ll have a fresh, lightweight VM ready to go!

Thanks!