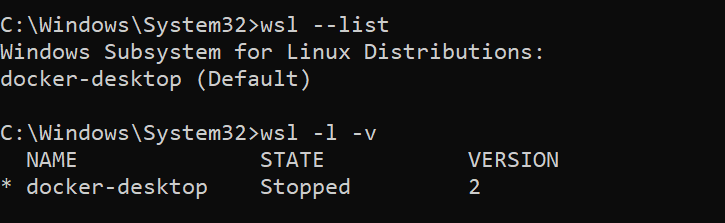
**Installing Ansible and Running a Playbook**

Need to have Python(3.8 ver and abv, I’m using Python3.12), WSL (Windows Subsystem for Linux) OR VMware where you can spin up an Ubuntu VM. Here, I’m using WSL:  
PS: When I installed Docker Desktop for Windows, WSL got installed with it

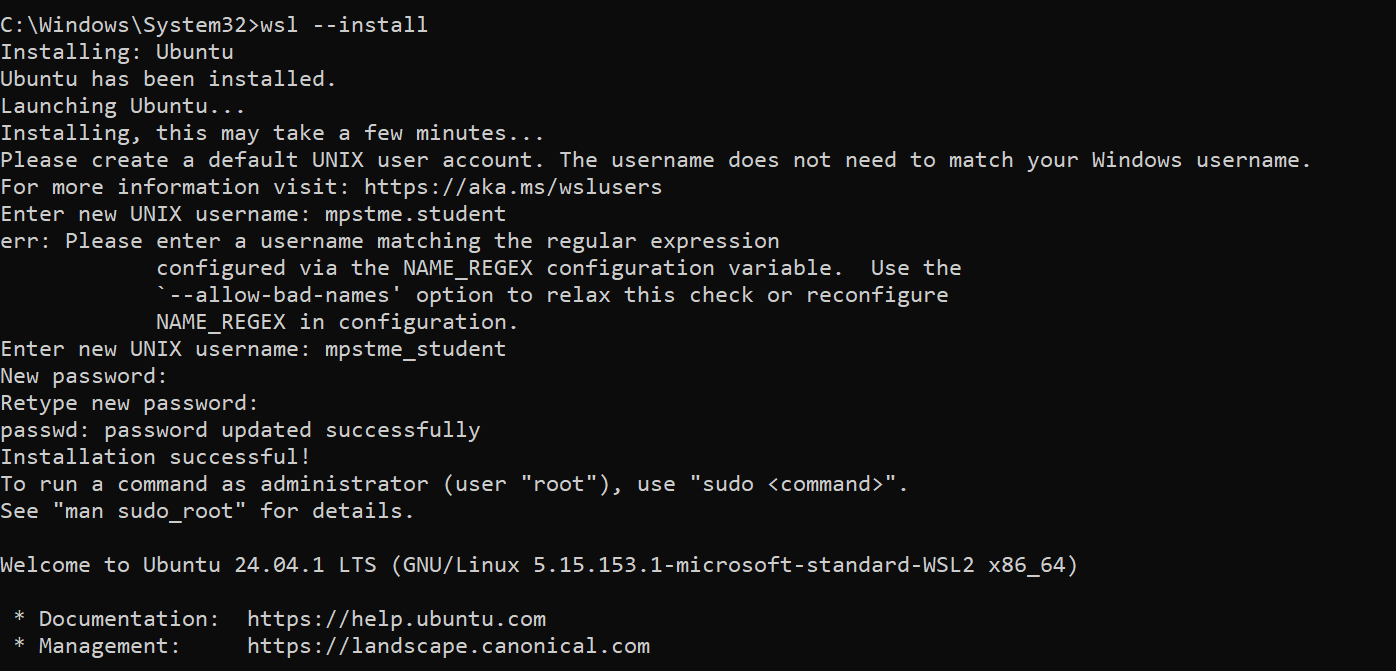
1. Click on Windows icon on your keyboard and type CMD. Click on Run As Administrator
2. The terminal of Windows will open and you should see the prompt C:/Windows/System32
3. Type wsl –list OR wsl -l -v to see the list of install WSL instances on your computer



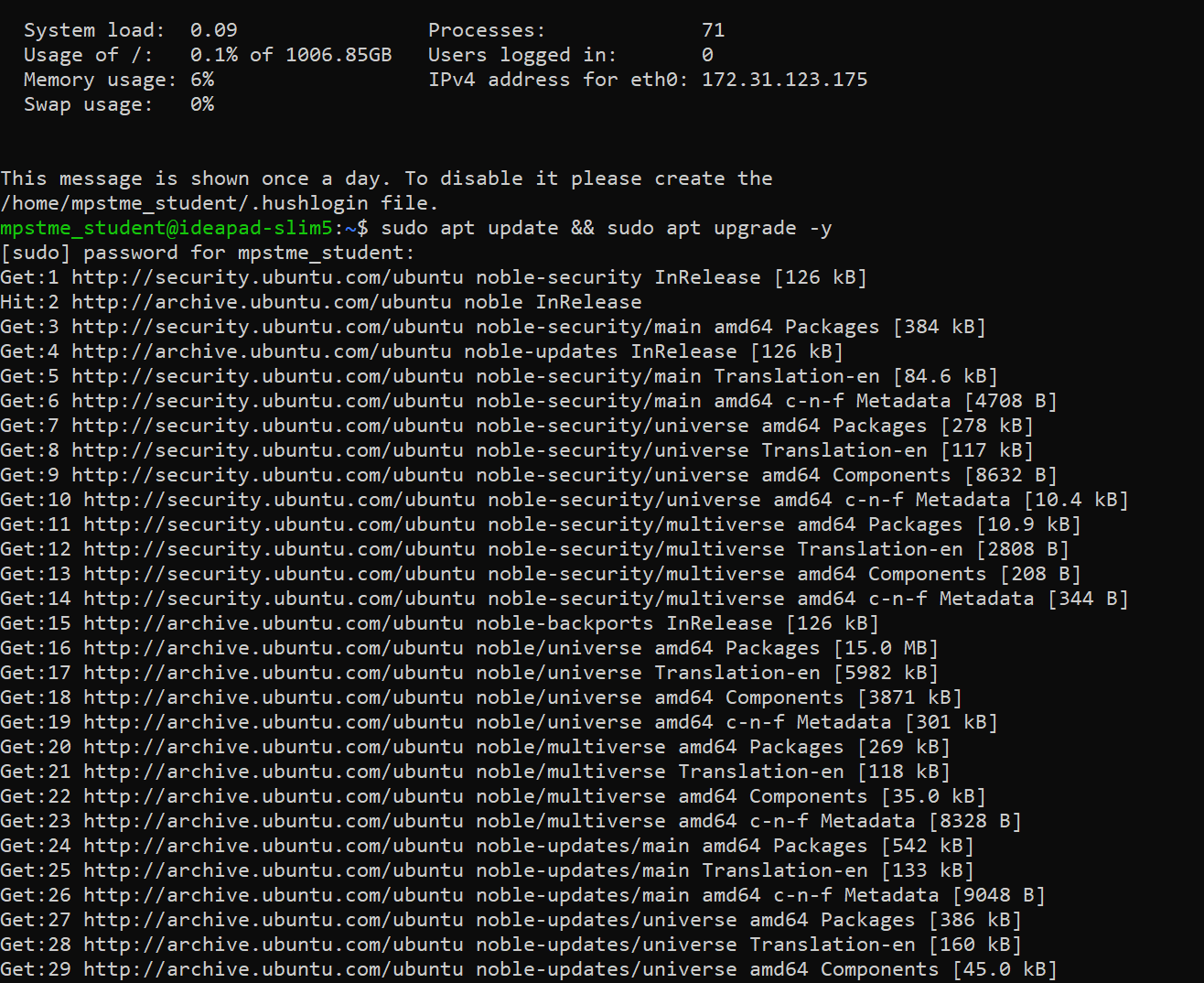
1. Then type wsl –install to install the Ubuntu Distribution of Linux
2. Give in a UNIX username and password

UNIX username: mpstme\_student

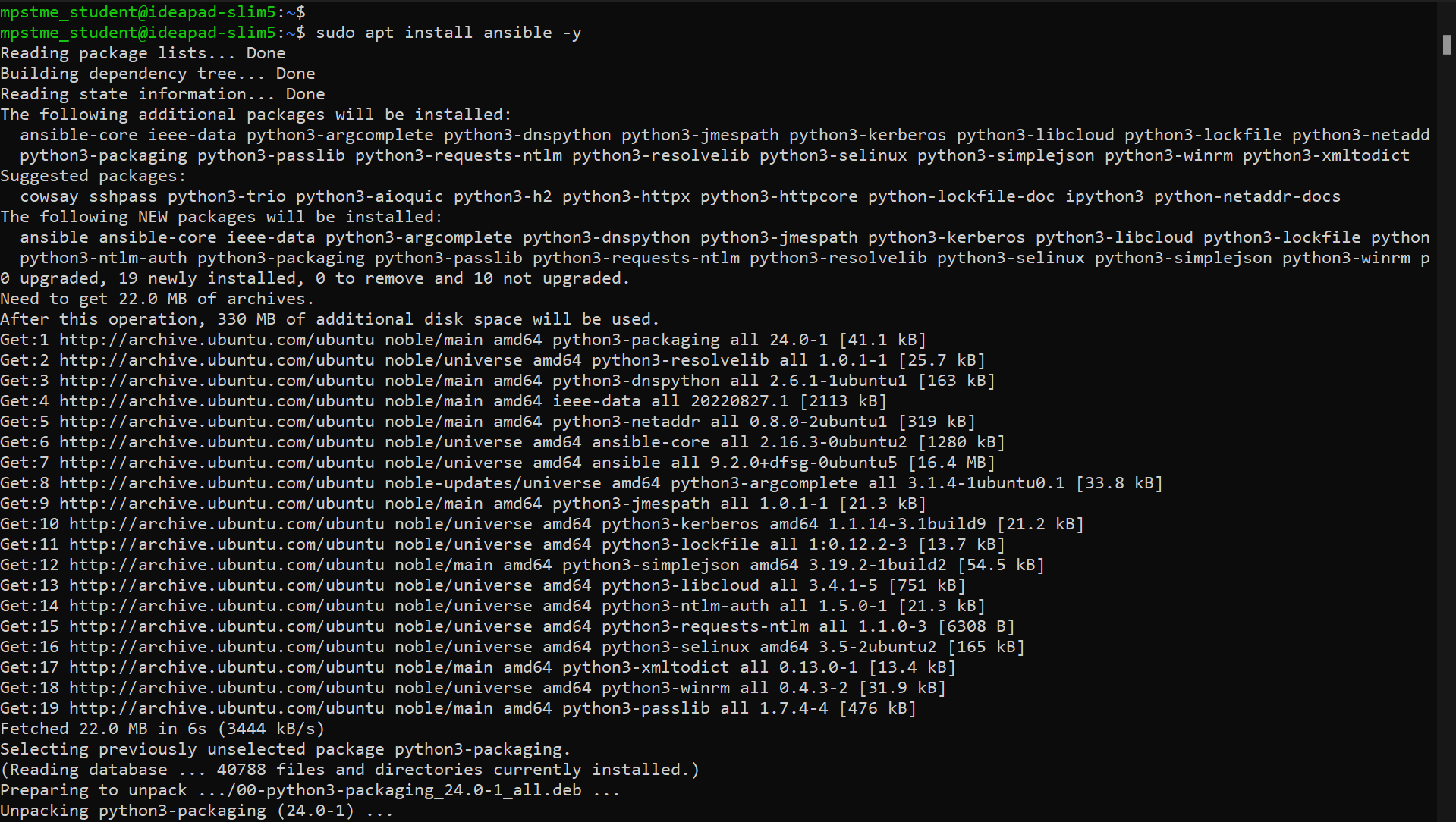
Password: renurao123



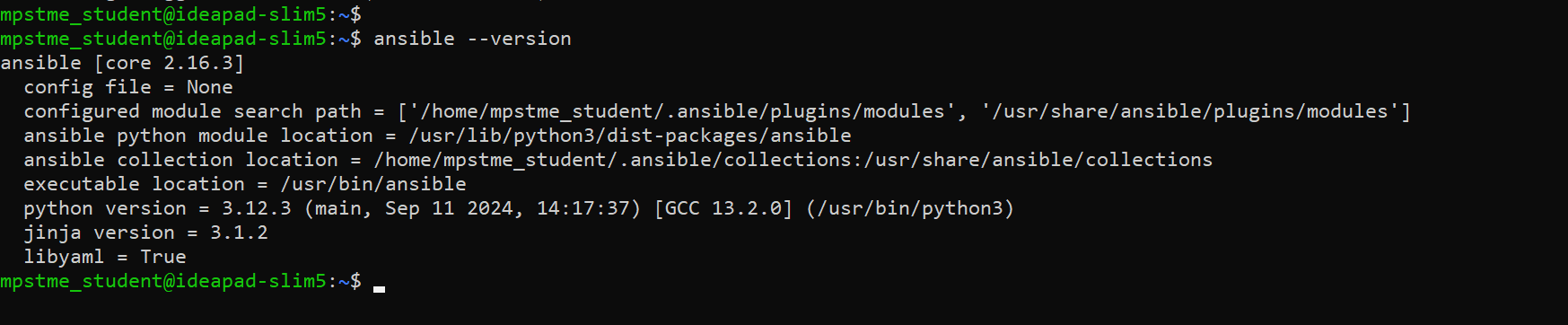
1. After the installation of Ubuntu is complete, type sudo apt update && sudo apt upgrade -y



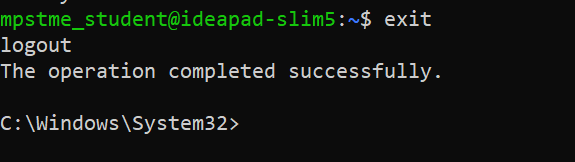
1. Next, we install ansible using: sudo apt install ansible -y



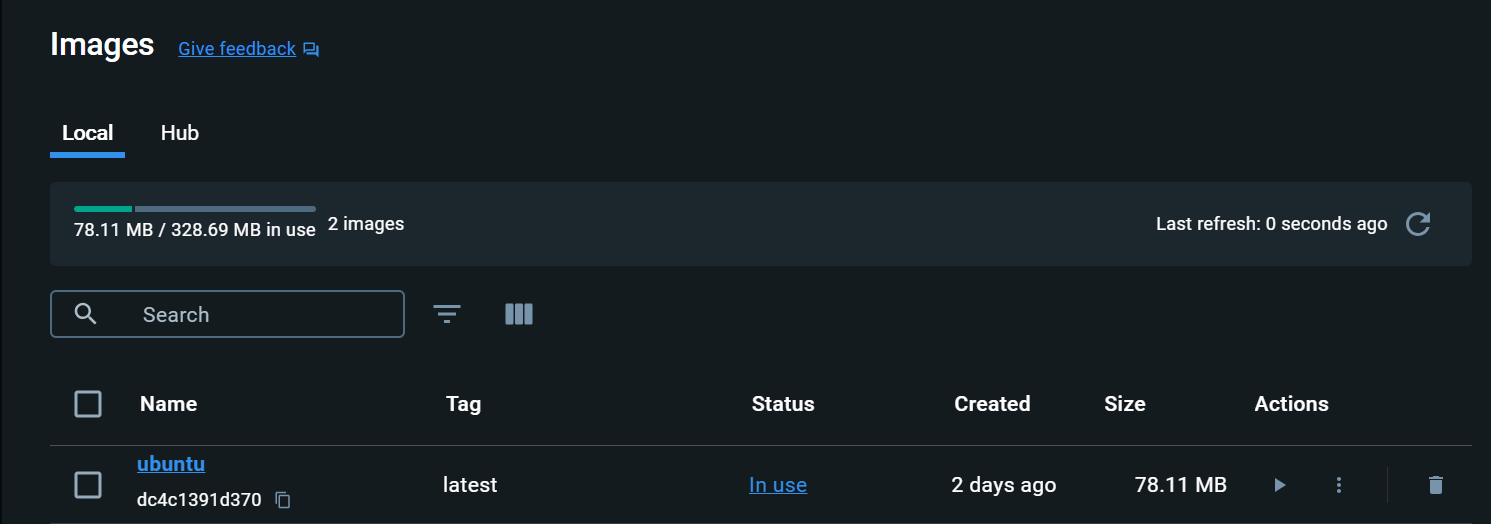
1. To check if ansible has been installed successfully: ansible –version



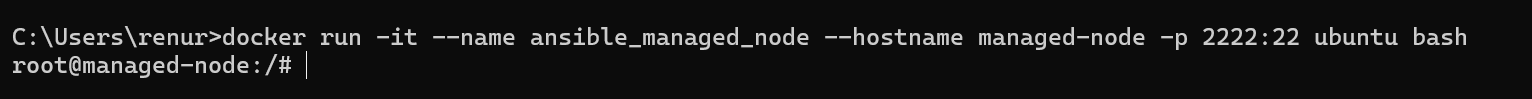
1. Now we’ve got ansible setup on WSL Ubuntu machine. This will be called as **Control Node**. Now we need to create **Managed Nodes** for which I’ll be using an Ubuntu Docker image to spin up a container and in that we install SSH and generate the keys for remote access, from WSL ubuntu machine to Ubuntu Docker container.
2. For logging out,



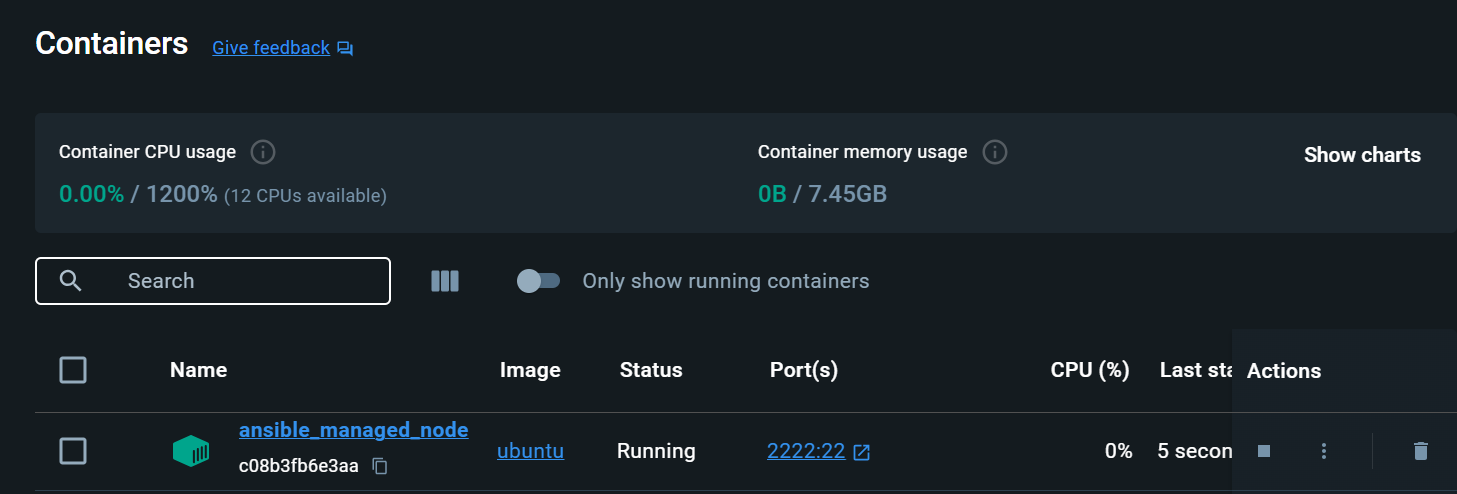
1. To create the Ubuntu container as a managed node, go to docker hub on google, search for ubuntu image and click on the docker official image of Ubuntu.
2. Copy the command and paste it in CMD: docker pull ubuntu
3. In Docker Desktop, you should be able to see the image of Ubuntu



1. Now spin up the container

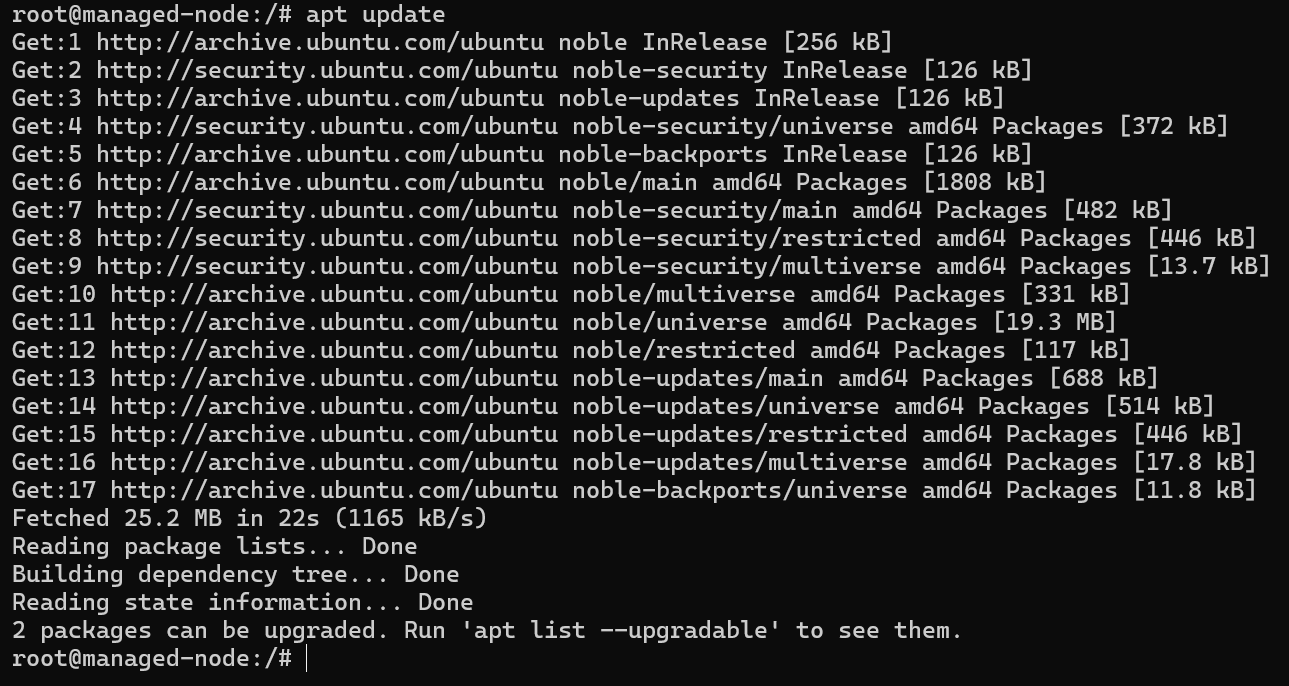


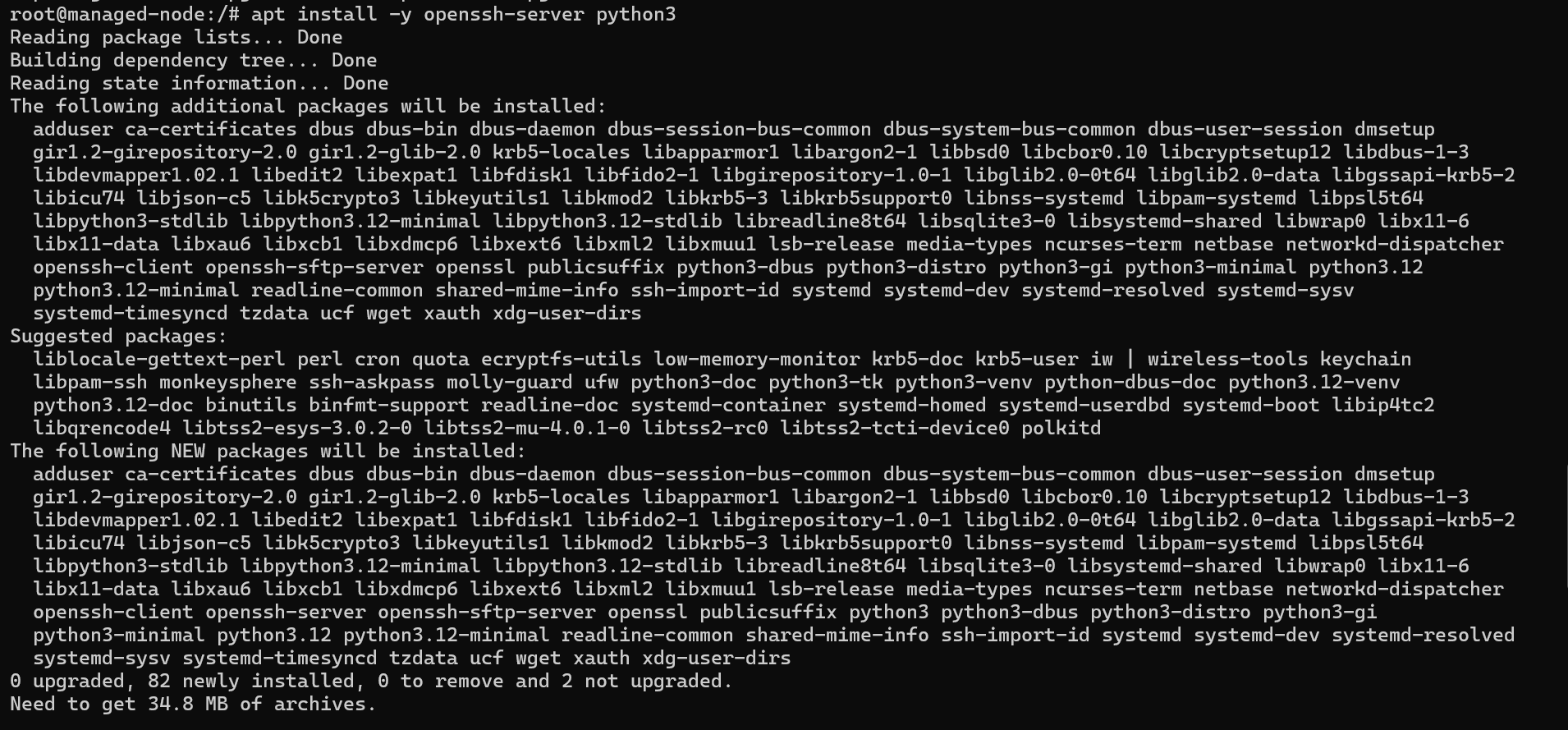
1. You should be able to see the container running in Docker Desktop



1. Now we’ve entered the bash (CMD) of the container. Now install SSH and Python3 on the container using this bash: apt update

apt install -y openssh-server python3

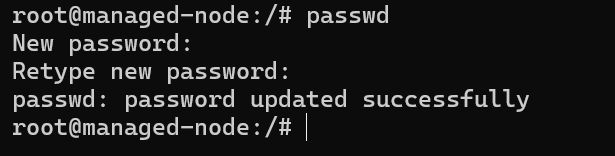




1. Now start SSH on the container

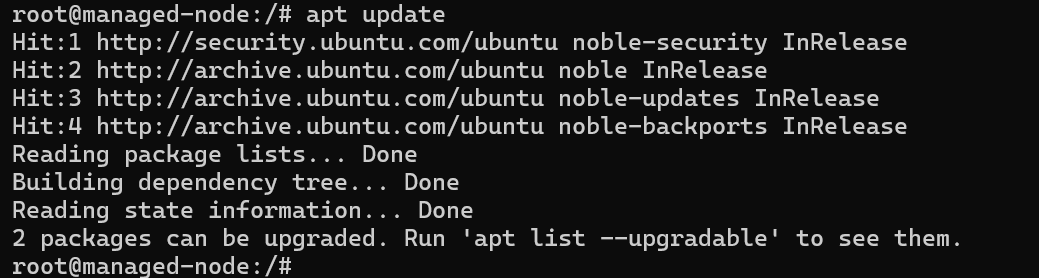


1. Now setup password for SSH on the container

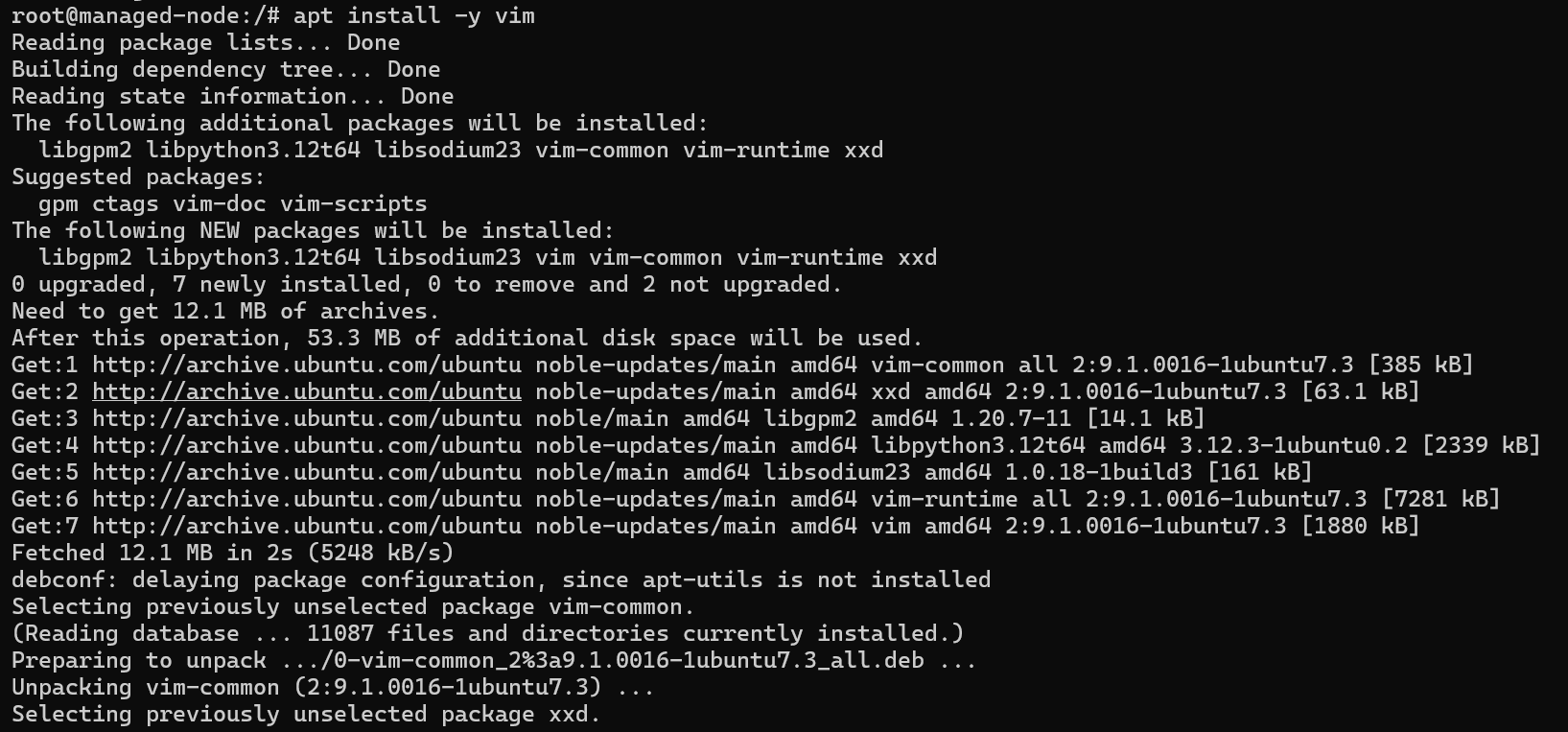


1. Now we install vim or vi on the container’s bash

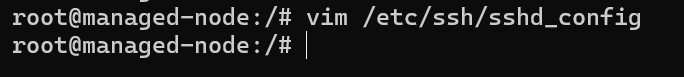
apt update

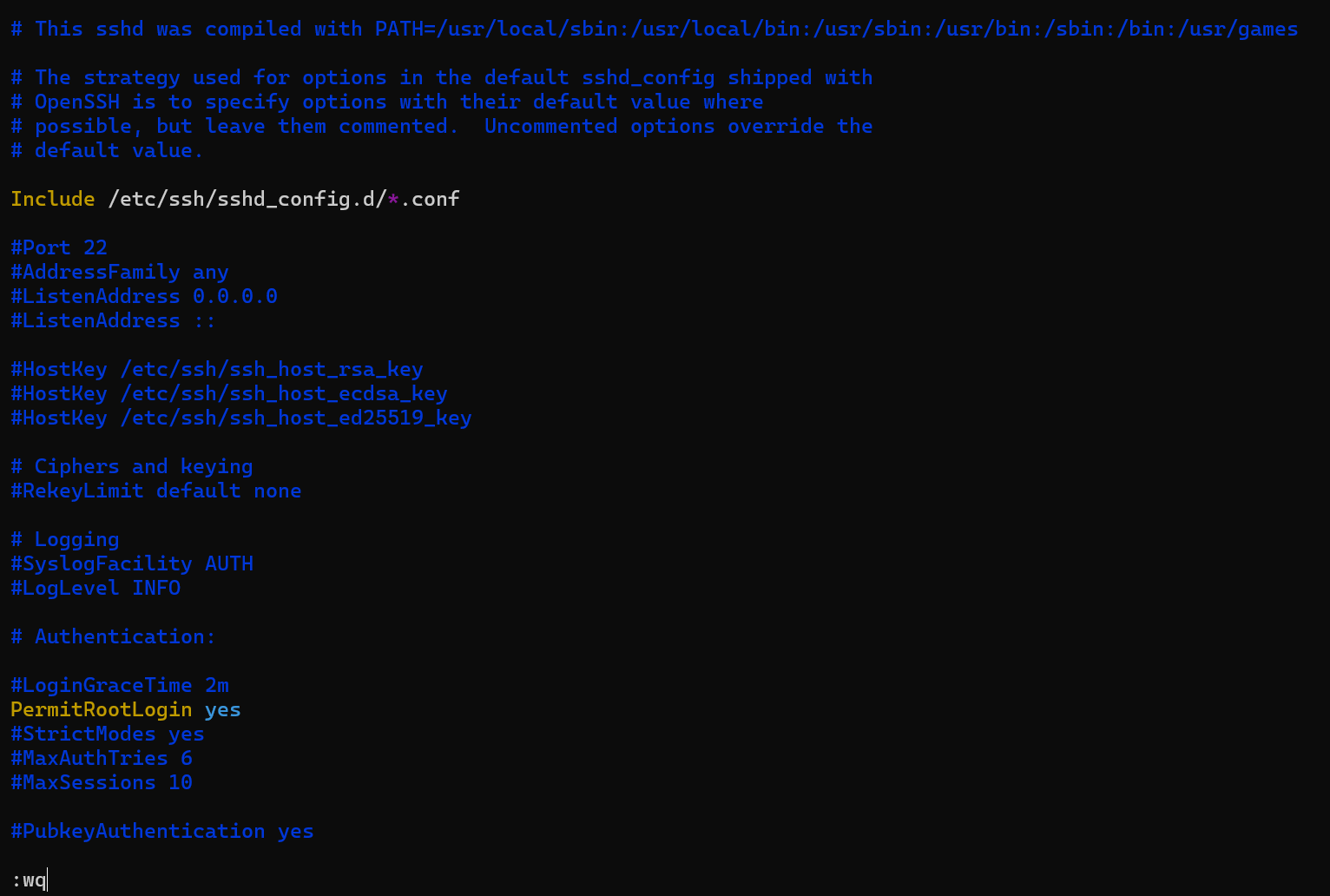


apt install -y vim



1. Edit /etc/ssh/sshd\_config file and save using :wq

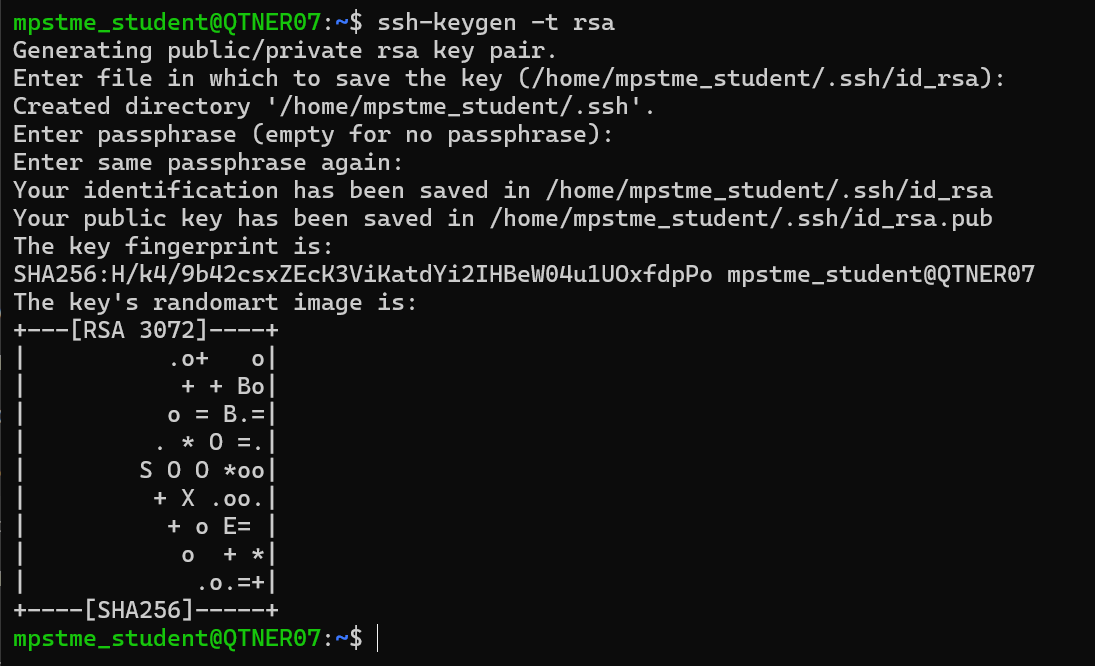




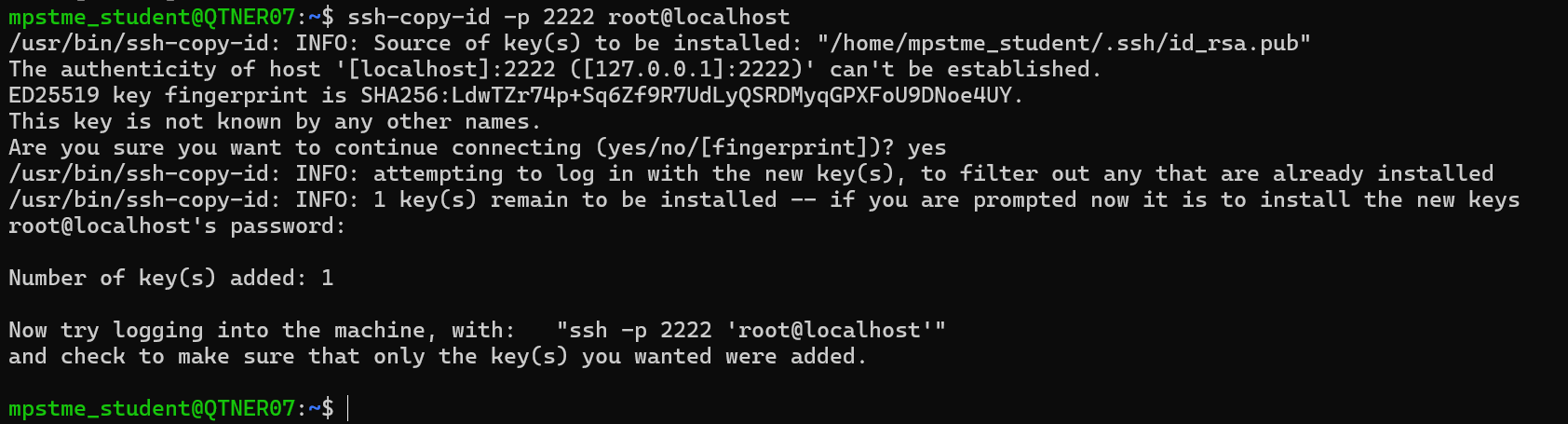
1. Restart SSH on the container



1. Now create SSH public key on WSL Ubuntu machine



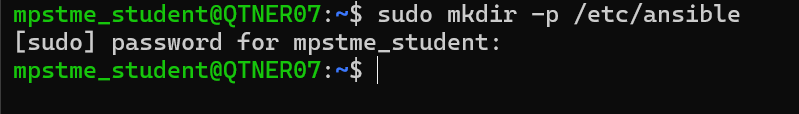
1. Copy the public key of SSH of WSL Ubuntu onto the Ubuntu container



This will add your public SSH key to the container’s authorized\_keys file, allowing Ansible to connect to the container without a password.

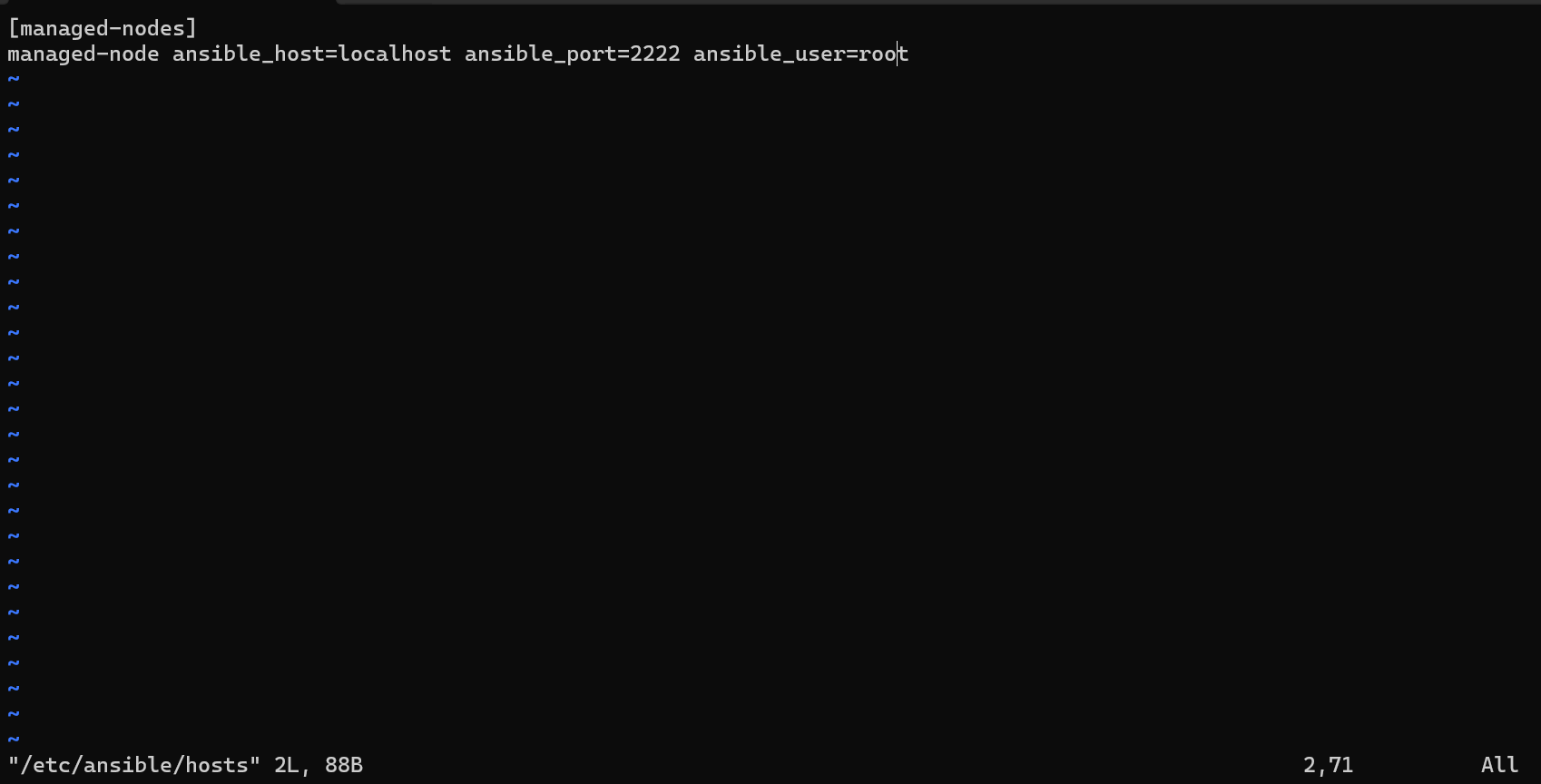
1. Configure an Ansible inventory on WSL Ubuntu, where you will specify the hosts that Ansible needs to manage:

For that first create this directory



1. Then open this file in sudo vim





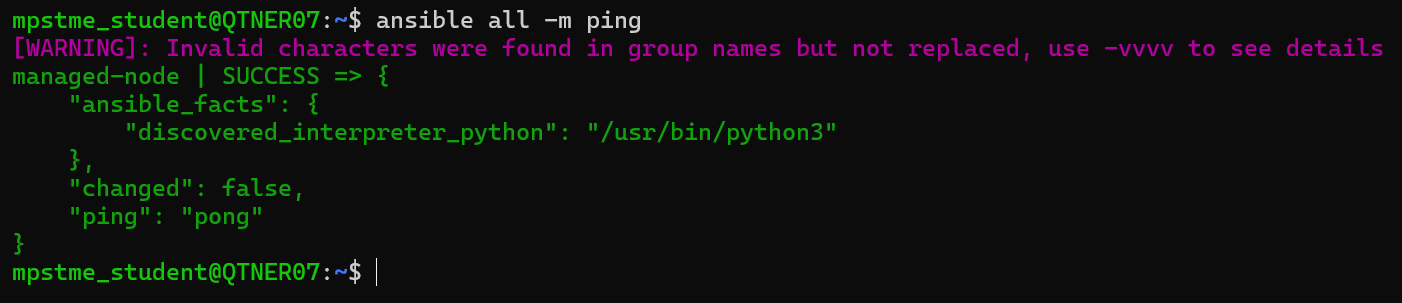
i to insert data

ESC

:wq to save and exit

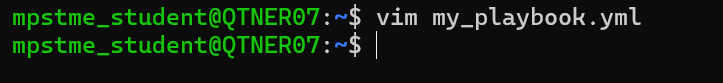
1. Test the connection from Ansible to your managed node:

ansible all -m ping

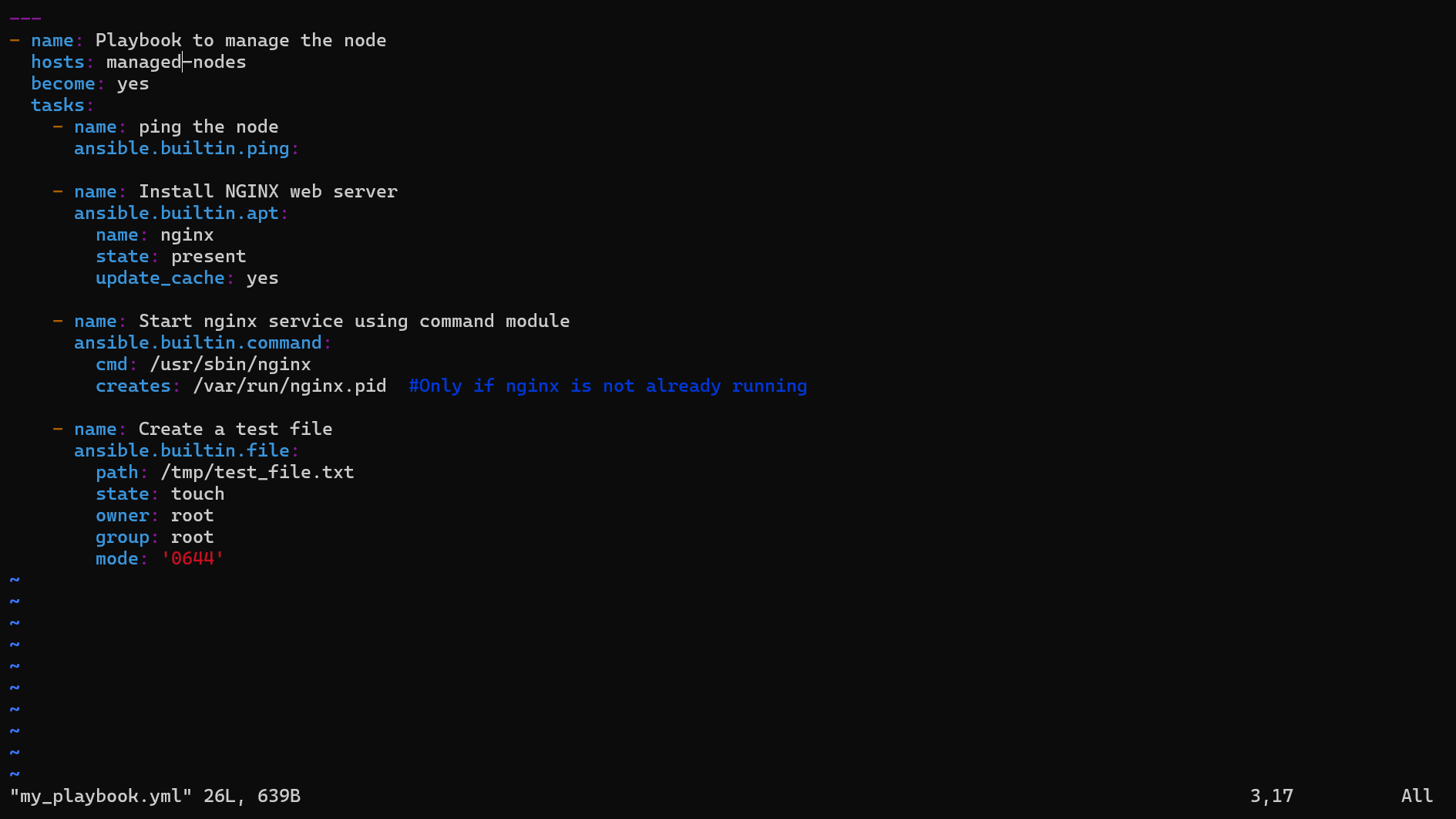


You should get the pong response

1. Now we will create our first playbook. Type vim my\_playbook.yml in WSL Ubuntu:

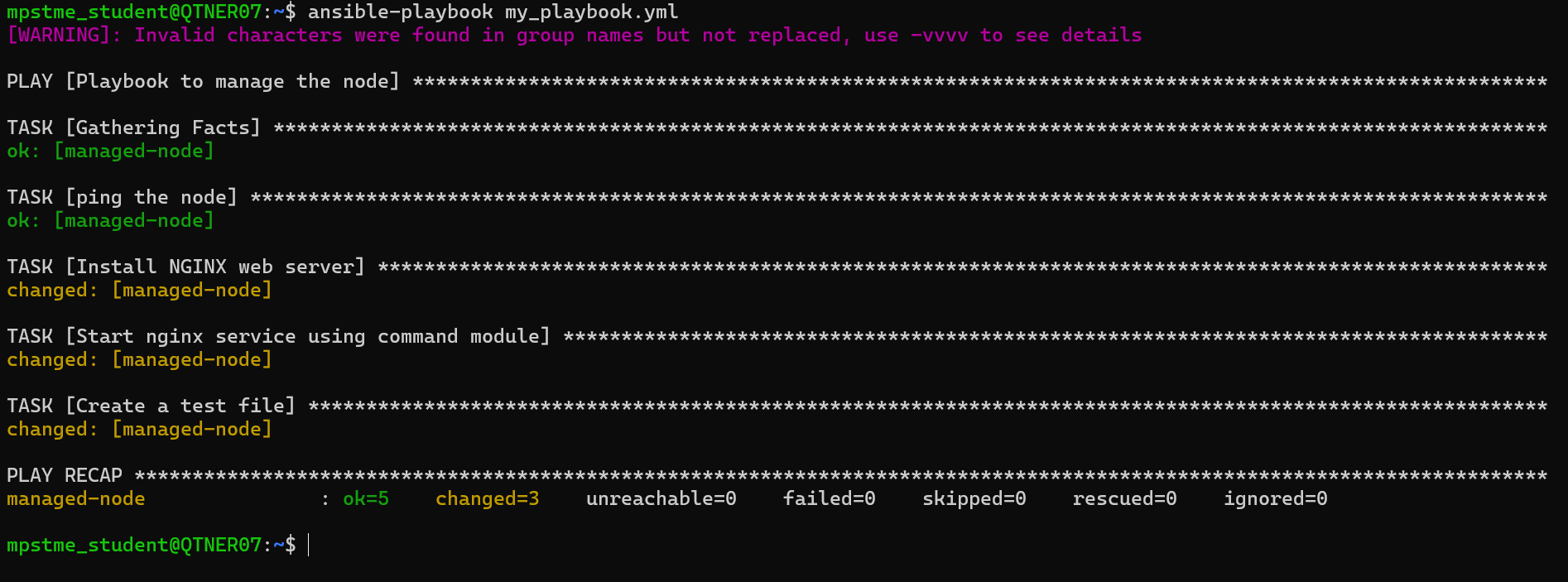


1. This is the content inside the playbook

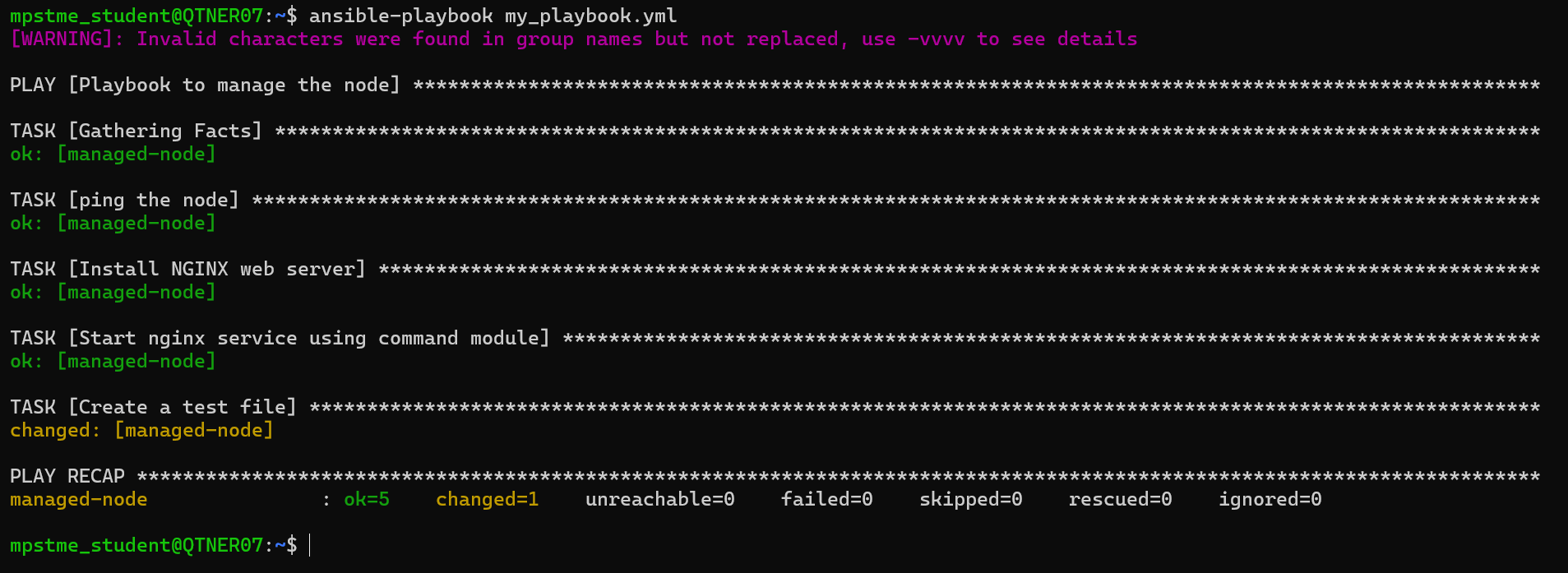


Press ESC and :wq

1. Now run the playbook using: ansible-playbook my\_playbook.yml



1. If you run the playbook again,



**Ansible With Apache**

---

- name: Install and ensure Apache is running

hosts: all

become: yes

tasks:

- name: Update the package repository

apt:

update\_cache: yes

- name: Install Apache

apt:

name: apache2

state: present

- name: Ensure Apache service is started using systemd

systemd:

name: apache2

state: started

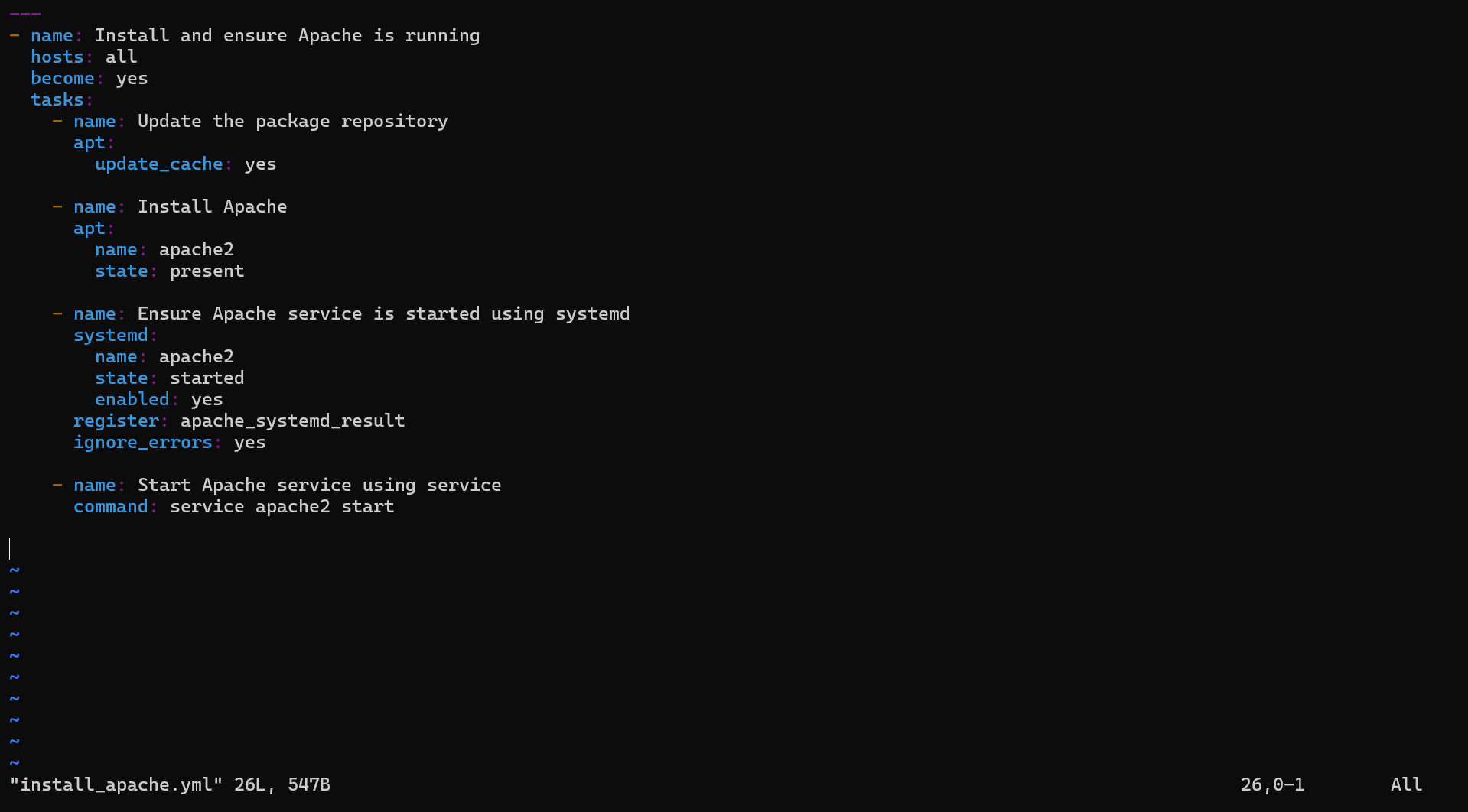
enabled: yes

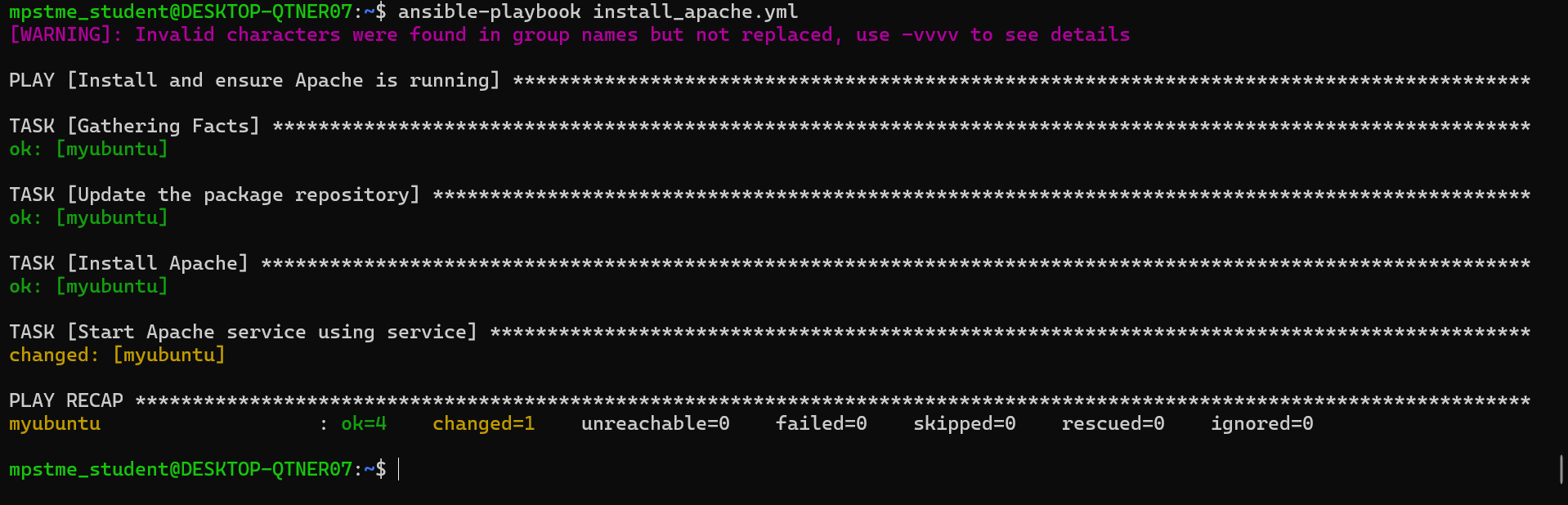
register: apache\_systemd\_result

ignore\_errors: yes

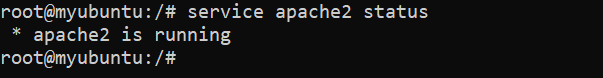
- name: Start Apache service using service

command: service apache2 start







****