**ANSIBLE**

* **INSTALLATION**

**Installing WSL on Windows:**

**Step 1: On Windows cmd, run command**

**wsl --install**

**This will install both WSL and Ubuntu**

**In settings go to**

**A white text on a black background

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Step 2: Reboot Your PC**

**Step 3: Create a new Ubuntu user. Enter Your Username and Password**

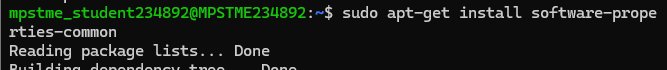
**A computer screen with white text

Description automatically generated**

**Step 4: Run sudo apt update- to install all updates**

****

**Step 5: Run the sudo apt-get install software-properties-common, to get all common properties**

****

**Step 6: Run sudo apt-add-repository ppa:ansible/ansible to install repository**

****

**Step 7: Run sudo apt update again to update the repository**

**A black screen with white text

Description automatically generated**

**Step 8: Run sudo apt-get install ansible**

**A screenshot of a phone

Description automatically generated**

**Step 9: Run ansible --version**

**A screenshot of a computer

Description automatically generated**

* **Creating Remote Servers Using Docker and installing SSH**

**Install docker, from docker hub pull the image**

**Step 1: Pull Docker image of ubuntu from docker hub and run**

**docker pull ubuntu –on cmd**

**A screen shot of a computer

Description automatically generated**

1. **In Docker Desktop, you should be able to see the image of Ubuntu**

**A screen shot of a computer

Description automatically generated**

1. **Now spin up the container:**

**Code: docker run -it --name abc --hostname managed-node -p 2222:22 ubuntu bash**

****

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

**A screenshot of a computer

Description automatically generated**

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**

**A screen shot of a computer

Description automatically generated**

**A black screen with white text

Description automatically generated**

1. **Now start SSH on the container**

**Code: service ssh start**

****

1. **Now setup password for SSH on the container**

**A black screen with white text

Description automatically generated**

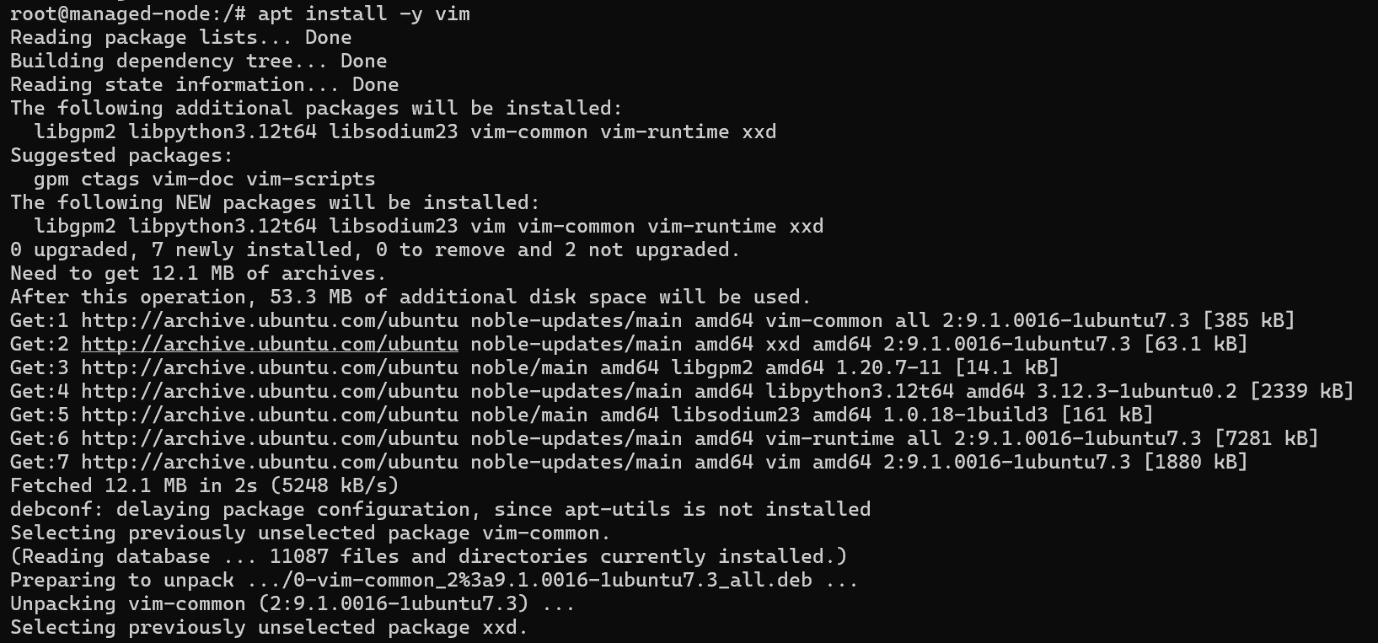
1. **Now we install vim or vi on the container’s bash**

**apt update**

**A computer screen shot of a program

Description automatically generated**

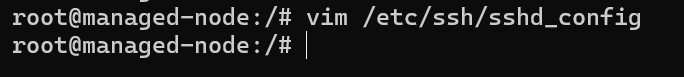
**apt install -y vim**

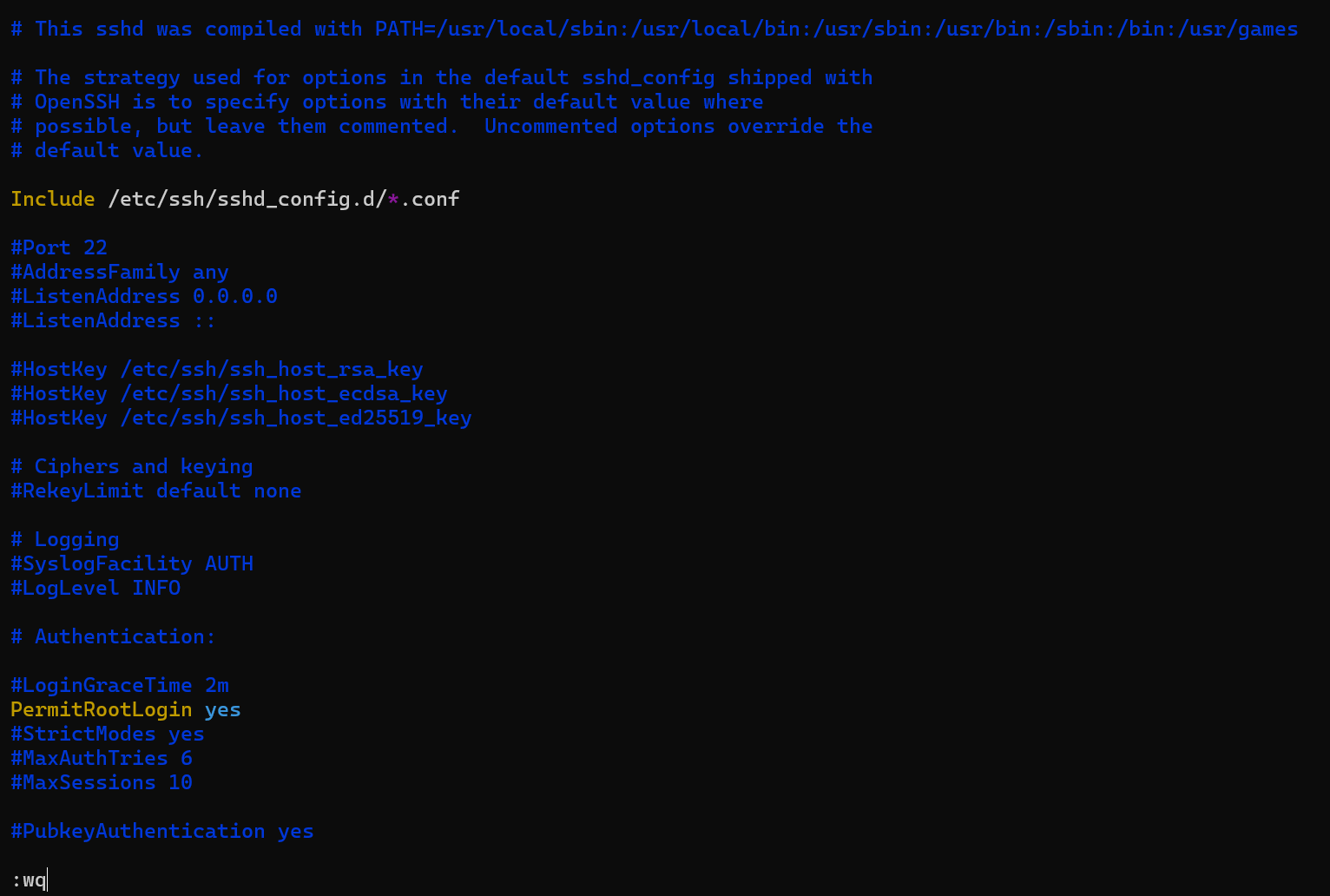
****

1. **Edit this file and save using :wq**

**Code: vim /etc/ssh/sshd\_config**

**Change permission root login to yes**

****

****

1. **Restart SSH on the container**

**Code: service ssh restart**

****

1. **On create SSH public key on WSL Ubuntu machine**

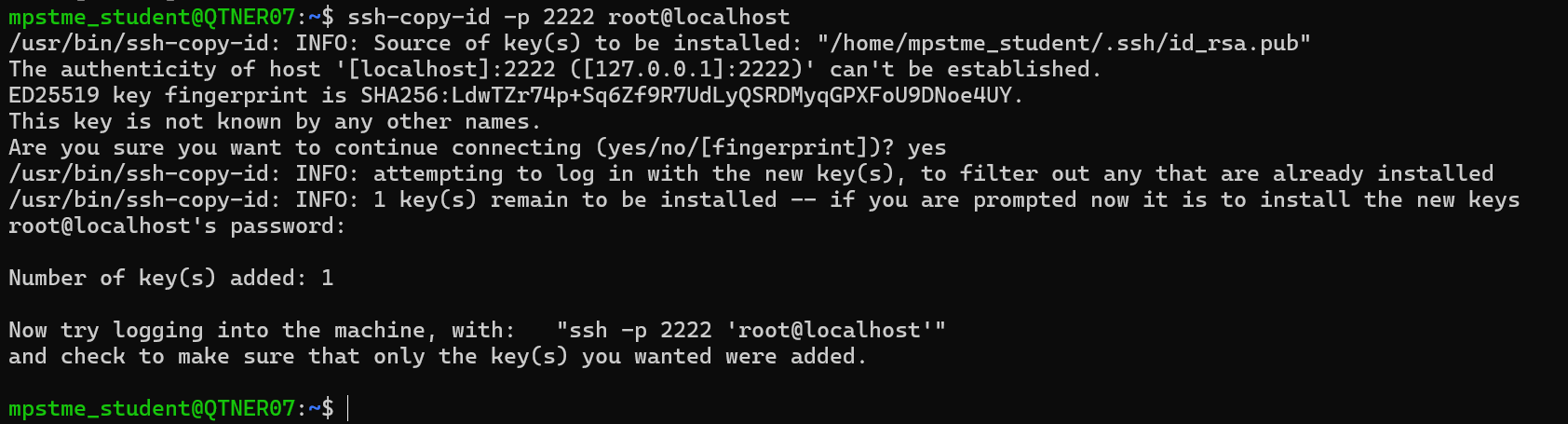
**Ssh-keygen -t rsa**

**A computer screen shot of a black screen

Description automatically generated**

**In ubunto**

**Code: ssh-copy-id -p 2222 root@localhost**

****

1. **Now add the manage nodes(docker codes) to the container**

**Code: sudo vi/etc/ansible/hosts**

****

1. **Open the hosts file and below the comments section add the two lines:**

**[managed\_nodes]**

**managed-node ansible\_host=localhost ansible\_port=2222 ansible\_user=root**

**A screenshot of a computer program

Description automatically generated**

**:wq to save and exit**

1. **Test your connections**

**Command: ansible all -m ping**

**A black screen with purple text

Description automatically generated**

* **Creating Playbooks**

**Step 1: Run mkdir playbooks, to create a new directory to store your playbooks**

**Step 2: Next run ls to verify**

**Code:ls**

****

**Step 3: Go inside the playbooks folder, run cd playbooks:**

**Cd playbooks**

****

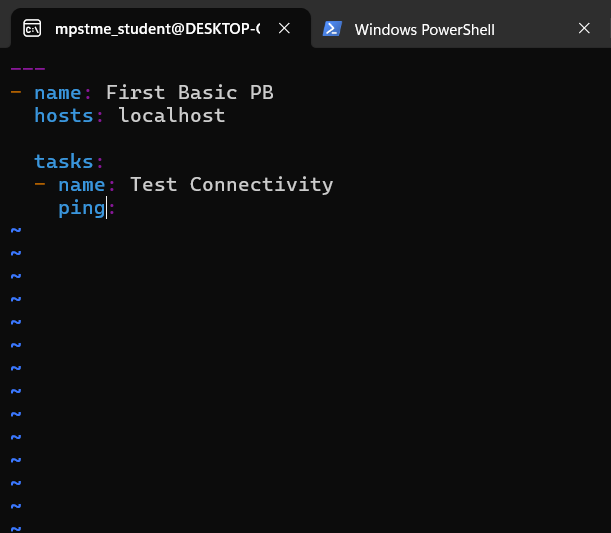
**Step 4:Create your playbook using command:**

**Code:** **vi first\_pb.yml**

****

**Step 5:Open vi editor and write your content**

* **Press “i” to start writing**

****

**Code:**

**---**

**- name: First Basic PB**

**hosts: localhost**

**tasks:**

**- name: Test Connectivity**

**ping:**

**To exit, press ESC and write :wq in the editor you will go back to your command shell**

**To see your file again, run**

****

**Step 6: To run your playbook, run command**

**Code: ansible-playbook first\_pb.yml**

**A screen shot of a computer program

Description automatically generated**

**Practice Tasks:**

**Task 1: Test Connectivity with localhost and printing a hello world**

**Vi File:**

**A screen shot of a computer program

Description automatically generated**

---

- name: First Basic PB

hosts: localhost

tasks:

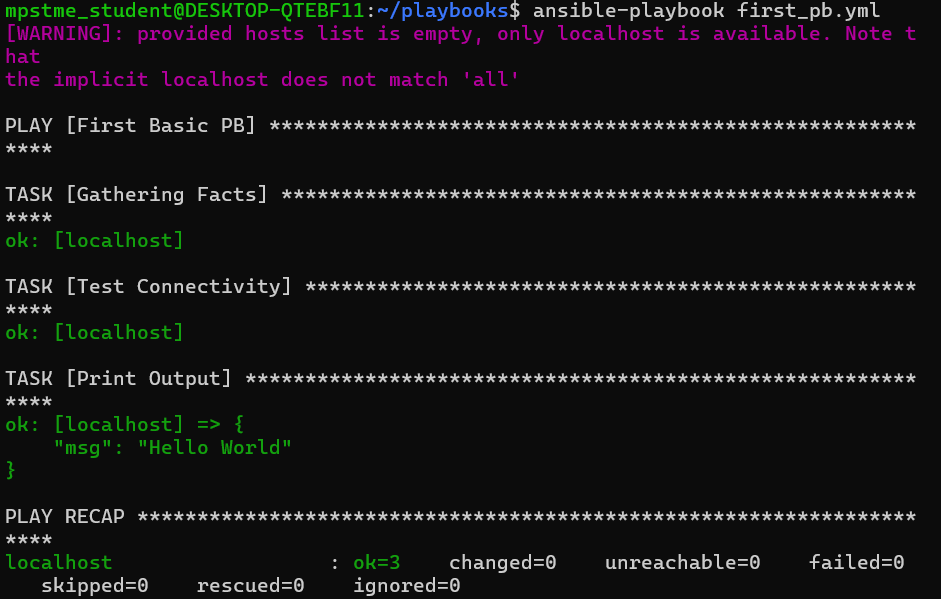
- name: Test Connectivity

ping:

- name: Print Output

debug: msg="Hello World"

**Running the task:**



Task 2: Installing a package on localhost

**Create a new playbook:**

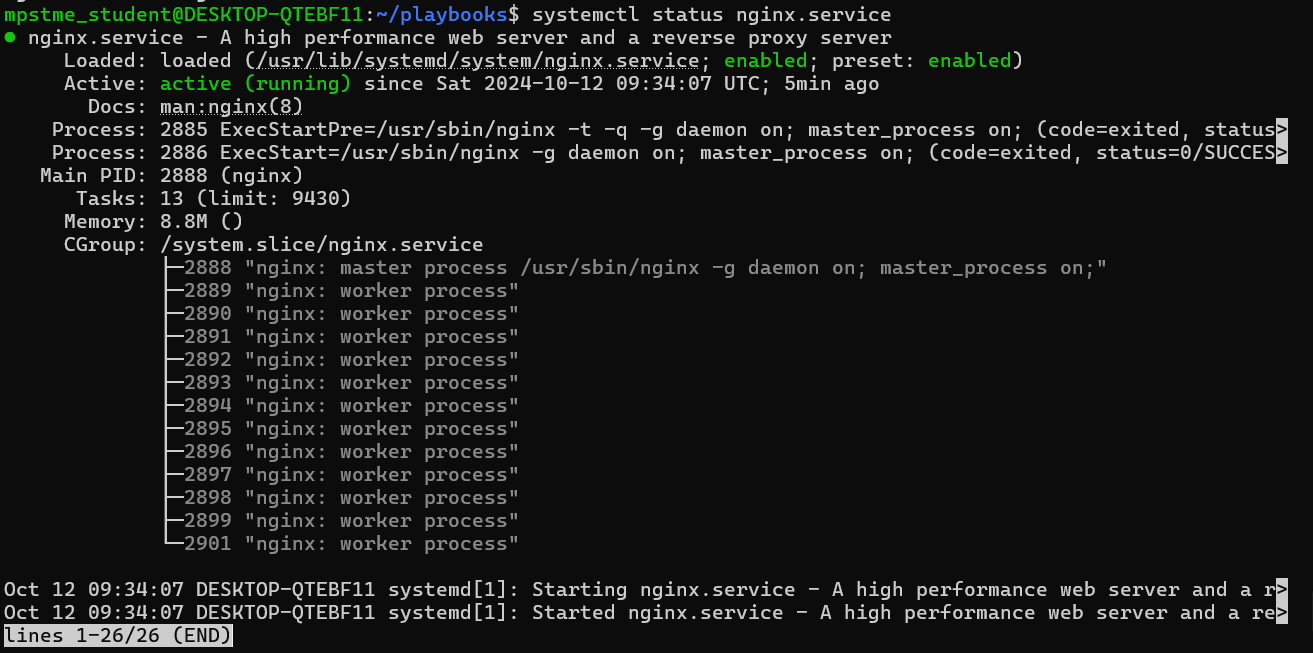
****

**Check if nginx Is installed successfully (TASK 1):**



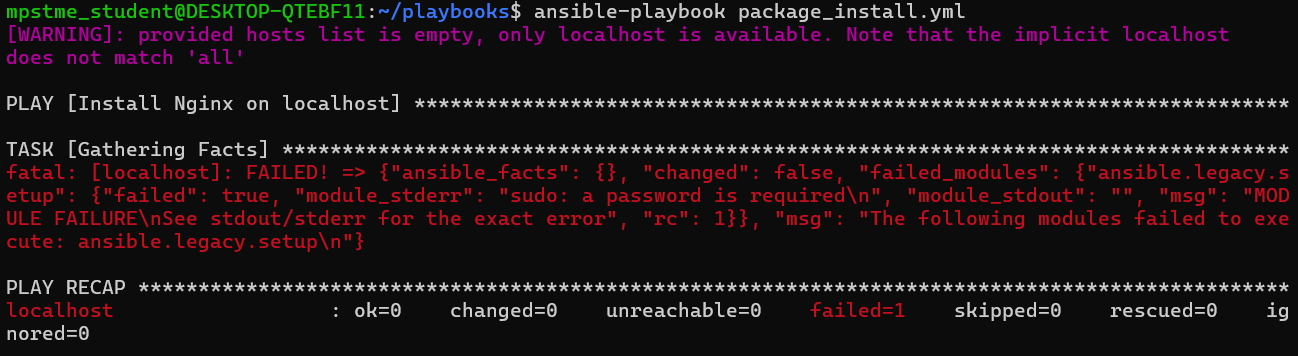
**TASK 2: If nginx service is running successfully:**

**systemctl status nginx.service**



**Errors:**

**1.**



**Resolution: use –ask-become-pass with the run command**

A screen shot of a computer

Description automatically generated

Creating a Playbook to run tasks on remote server

A screen shot of a computer program

Description automatically generated

---

- name: Simple playbook to manage the node

hosts: managed\_nodes

become: yes # Escalate privilege to root

tasks:

- name: Ping the node

ansible.builtin.ping:

- name: Install nginx web server

ansible.builtin.apt:

name: nginx

state: present

update\_cache: yes

- name: Start nginx service using command module

ansible.builtin.command:

cmd: /usr/sbin/nginx # Run nginx in the foreground

creates: /var/run/nginx.pid # Only run if nginx is not already running

- name: Create a test file

ansible.builtin.file:

path: /tmp/test\_file.txt

state: touch

owner: root

group: root

mode: '0644'

Run the file:



**For Apache:**

**Playbook:**

---

- name: Install and ensure Apache is running

hosts: localhost

become: yes # Run tasks as sudo

tasks:

- name: Update the package repository

apt:

update\_cache: yes

- name: Install Apache

apt:

name: apache2

state: present

- name: Ensure Apache service is started and enabled

service:

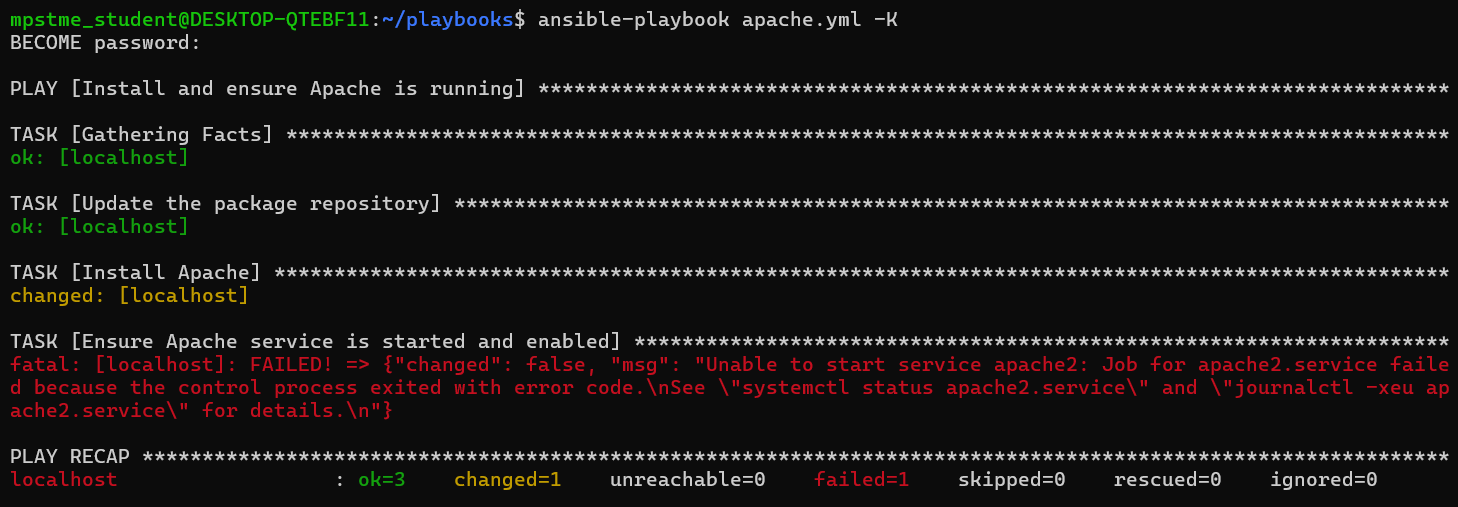
name: apache2

state: started

enabled: yes

**A black screen with white text

Description automatically generated**

****

**A screenshot of a computer

Description automatically generated**

**A black screen with white text

Description automatically generated**