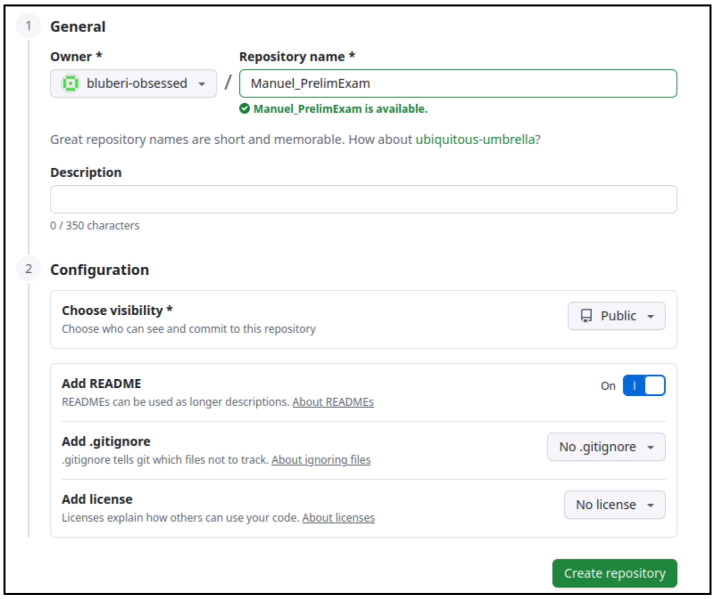
Hands-on Prelim Exam	
Course Code: CPE 212	**Program:** BSCPE
Course Title: AUTOMATING SERVER MANAGEMENT	**Date Performed:** 08/29/2025
Section: CPE31S4	**Date Submitted:** 08/29/2025
Name: Hazel Aillson T. Manuel	**Instructor:** Engr. Robin Valenzuel
OUTPUT:	

1. Create a repository in your GitHub account and label it as Surname_PrelimExam



1 General

Owner * bluberi-obsessed / Repository name * Manuel_PrelimExam

Manuel_PrelimExam is available.

Great repository names are short and memorable. How about ubiquitous-umbrella?

Description

0 / 350 characters

2 Configuration

Choose visibility * Public
Choose who can see and commit to this repository

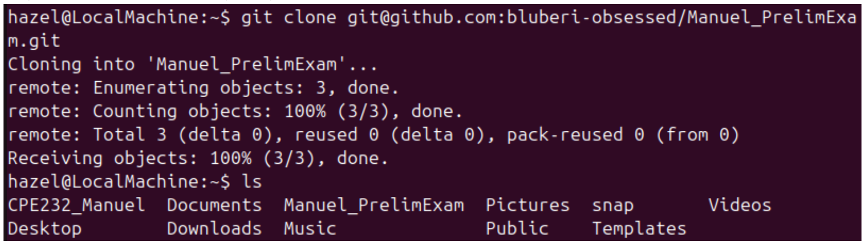
Add README On
READMEs can be used as longer descriptions. About READMEs

Add .gitignore No .gitignore
.gitignore tells git which files not to track. About ignoring files

Add license No license
Licenses explain how others can use your code. About licenses

Creating a new repository in Github. All of our files will be saved in this repository and this repository will be cloned to our Local machine.

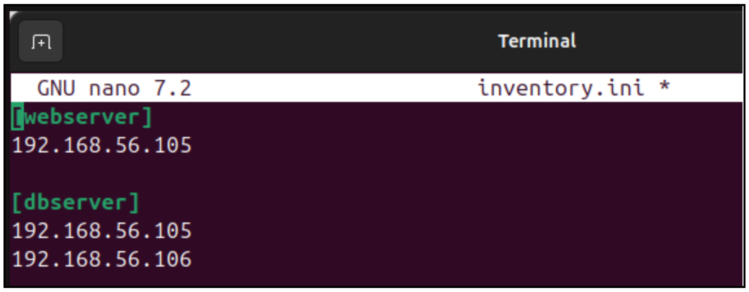
2. Clone your new repository in your CN.



```
hazel@LocalMachine:~$ git clone git@github.com:bluberi-obsessed/Manuel_PrelimExa
m.git
Cloning into 'Manuel_PrelimExam'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
hazel@LocalMachine:~$ ls
CPE232_Manuel  Documents  Manuel_PrelimExam  Pictures  snap       Videos
Desktop        Downloads  Music              Public    Templates
```

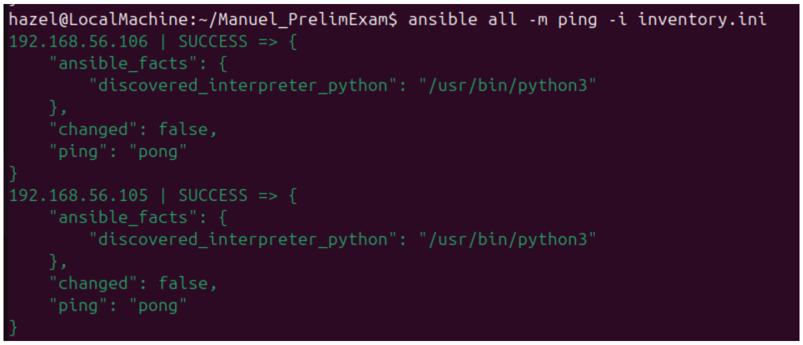
Cloning the repository file from Github to the local machine. After cloning, I checked the list of the current directory to confirm whether the repository is indeed cloned. As you can see, a new directory called "Manuel_PrelimExam" is in the directory.

3. In your CN, create an inventory file and ansible.cfg files.



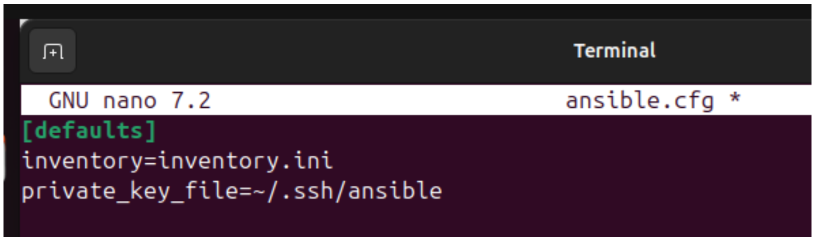
```
[webserver]
192.168.56.105

[dbserver]
192.168.56.105
192.168.56.106
```

Creating the inventory file that will store all of the servers that we had. We create two servers in this part, a webserver and a database one.



```
hazel@LocalMachine:~/Manuel_PrelimExam$ ansible all -m ping -i inventory.ini
192.168.56.106 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
192.168.56.105 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

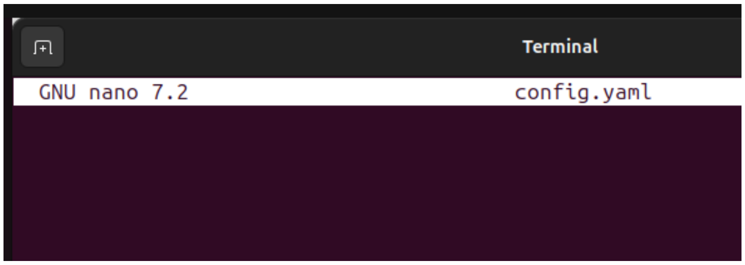
Confirming that there is no error with the inventory file through pinging. As you can see, all of the servers are reachable.



```
[defaults]
inventory=inventory.ini
private_key_file=~/.ssh/ansible
```

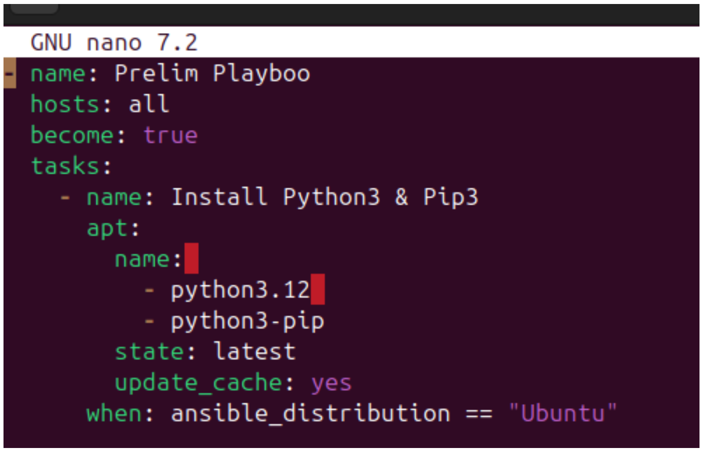
Creating the ansible configuration file to set up the defaults which includes the inventory file and private key.

4. Create an Ansible playbook that does the following with an input of a config.yaml file for both Manage Nodes

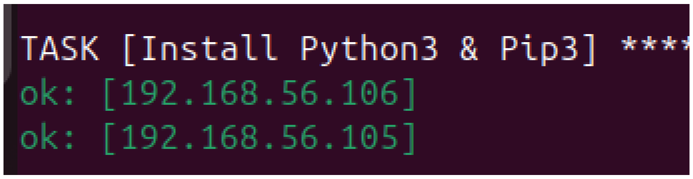


Creating the playbook file which will do the tasks below this.

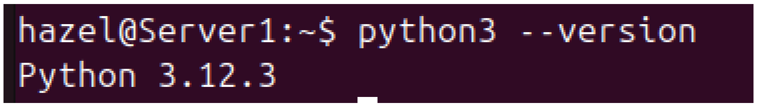
- Installs the latest python3 and pip3



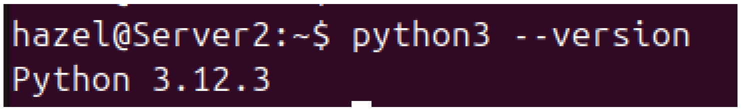
```
GNU nano 7.2
- name: Prelim Playboo
  hosts: all
  become: true
  tasks:
    - name: Install Python3 & Pip3
      apt:
        name:
          - python3.12
          - python3-pip
        state: latest
        update_cache: yes
      when: ansible_distribution == "Ubuntu"
```



```
TASK [Install Python3 & Pip3] ****
ok: [192.168.56.106]
ok: [192.168.56.105]
```



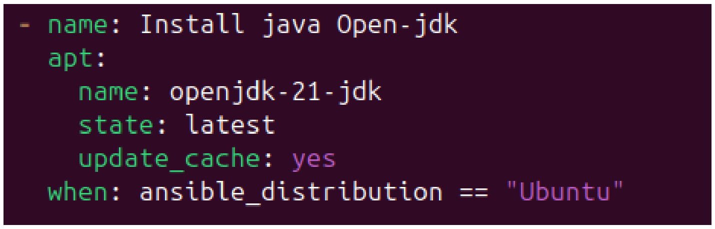
```
hazel@Server1:~$ python3 --version
Python 3.12.3
```



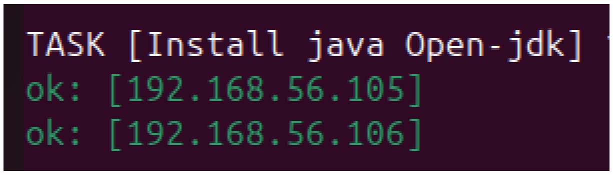
```
hazel@Server2:~$ python3 --version
Python 3.12.3
```

Creating an ansible playbook to install both python 3 and python-pip. In addition, images showing the version of the python are shown. Both of which are run from both Manage Nodes

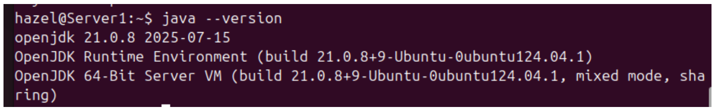
- use pip3 as default pip
- use python3 as default python
- Install Java open-jdk



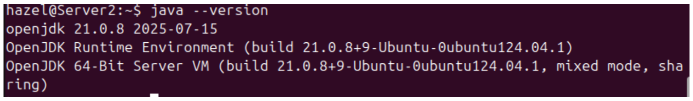
```
- name: Install java Open-jdk
  apt:
    name: openjdk-21-jdk
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"
```



```
TASK [Install java Open-jdk]
ok: [192.168.56.105]
ok: [192.168.56.106]
```



```
hazel@Server1:~$ java --version
openjdk 21.0.8 2025-07-15
OpenJDK Runtime Environment (build 21.0.8+9-Ubuntu-0ubuntu124.04.1)
OpenJDK 64-Bit Server VM (build 21.0.8+9-Ubuntu-0ubuntu124.04.1, mixed mode, sha
ring)
```

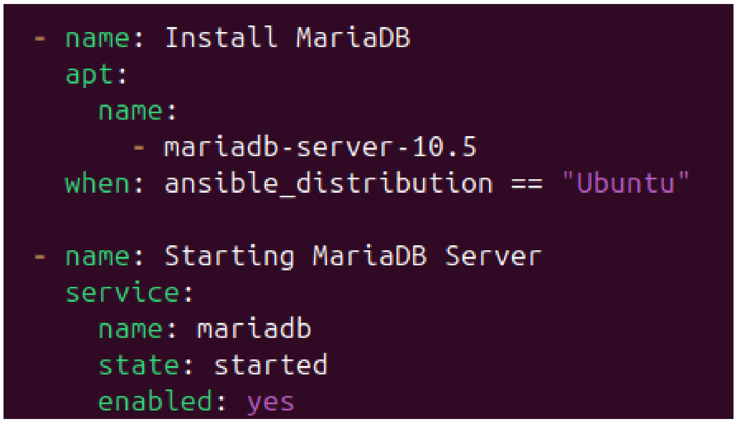


```
hazel@Server2:~$ java --version
openjdk 21.0.8 2025-07-15
OpenJDK Runtime Environment (build 21.0.8+9-Ubuntu-0ubuntu124.04.1)
OpenJDK 64-Bit Server VM (build 21.0.8+9-Ubuntu-0ubuntu124.04.1, mixed mode, sha
ring)
```

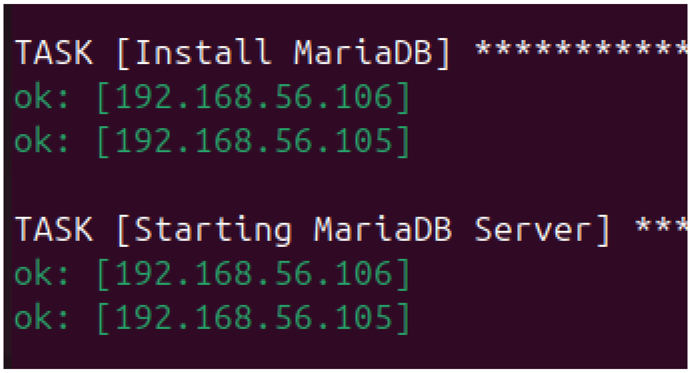
Installing Java Open jdk with Ansible. Furthermore, an additional image was

shown to confirm that the open-jdk was both installed to the server.

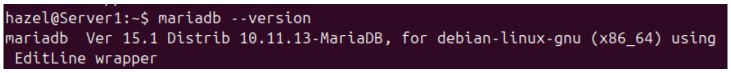
- Install MariaDB as well as starting the server, create a database and a table using mariaDB and input one record into a table USING ANSIBLE ONLY



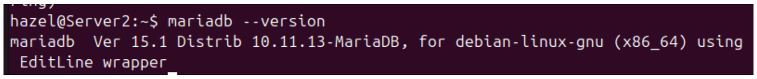
```
- name: Install MariaDB
  apt:
    name:
      - mariadb-server-10.5
  when: ansible_distribution == "Ubuntu"

- name: Starting MariaDB Server
  service:
    name: mariadb
    state: started
    enabled: yes
```



```
TASK [Install MariaDB] ***********
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [Starting MariaDB Server] ***
ok: [192.168.56.106]
ok: [192.168.56.105]
```

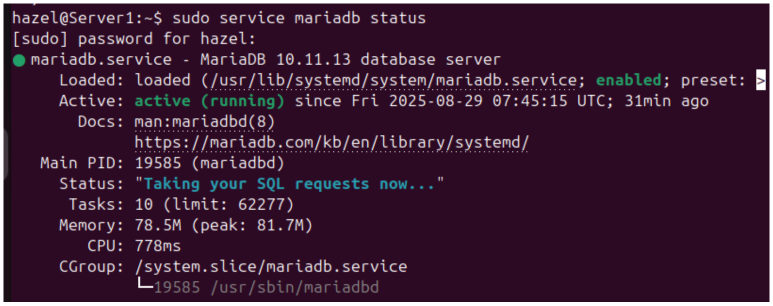


```
hazel@Server1:~$ mariadb --version
mariadb  Ver 15.1 Distrib 10.11.13-MariaDB, for debian-linux-gnu (x86_64) using
 EditLine wrapper
```

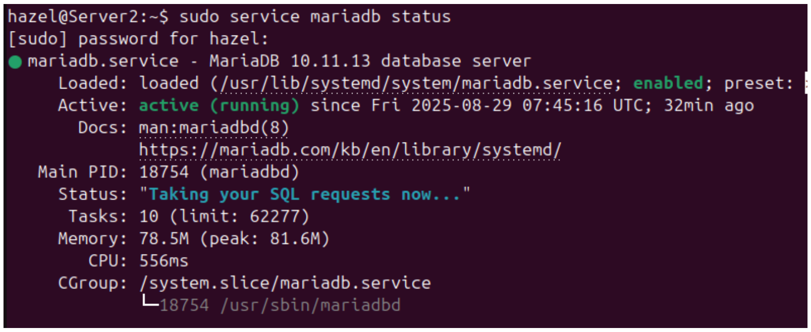


```
hazel@Server2:~$ mariadb --version
mariadb  Ver 15.1 Distrib 10.11.13-MariaDB, for debian-linux-gnu (x86_64) using
 EditLine wrapper
```

Just like the others, we installed the mariadb to both servers only using the local machine. Another way to actually show this is mysql –version.



```
hazel@Server1:~$ sudo service mariadb status
[sudo] password for hazel:
● mariadb.service - MariaDB 10.11.13 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: >
     Active: active (running) since Fri 2025-08-29 07:45:15 UTC; 31min ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 19585 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 62277)
     Memory: 78.5M (peak: 81.7M)
        CPU: 778ms
     CGroup: /system.slice/mariadb.service
             └─19585 /usr/sbin/mariadbd
```



```
hazel@Server2:~$ sudo service mariadb status
[sudo] password for hazel:
● mariadb.service - MariaDB 10.11.13 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset:
     Active: active (running) since Fri 2025-08-29 07:45:16 UTC; 32min ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 18754 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 62277)
     Memory: 78.5M (peak: 81.6M)
        CPU: 556ms
     CGroup: /system.slice/mariadb.service
             └─18754 /usr/sbin/mariadbd
```

Another thing we did is to start the server. As shown in the picture, both servers have an active and enabled mariadb server.

- Create Motd containing the text defined by a variable defined in config.yaml file and if there is no variable input the default motd is "Ansible Managed node by (your user name)"
- Create a user with a variable defined in config.yaml

5. PUSH and COMMIT your PrelimExam in your GitHub repo

https://github.com/bluberi-obsessed/Manuel_PrelimExam.git

git@github.com:bluberi-obsessed/Manuel_PrelimExam.git