Bernice M. Peña **CPE31S5**
Managing Enterprise Servers **Engr. Roman Richard**

Hands-on Prelim Exam

Tools Needed:

1. Control Node (CN) – 1
2. Manage Node (MN) - 1 Ubuntu
3. Manage Node (MN) - 1 CentOS

Procedure:

1. Note: You are required to create a document report of the steps you will do for this exam. All screenshots should be labeled and explained properly.
2. Create a repository in your GitHub account and label it as Surname_PrelimExam
3. Clone your new repository in your CN.
4. In your CN, create an inventory file and ansible.cfg files.
5. Create an Ansible playbook that does the following with an input of a config.yaml file for both Manage Nodes

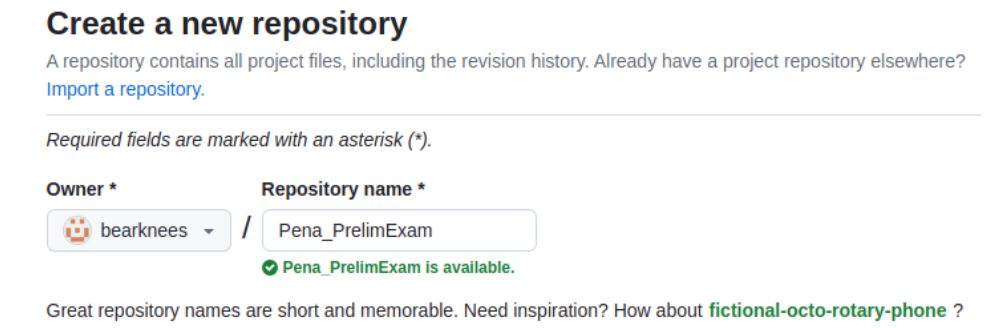
    - Installs the latest python3 and pip3
    - use pip3 as default pip
    - use python3 as default python
    - Install Java open-jdk
    - Create Motd containing the text defined by a variable defined in config.yaml file and if there is no variable input the default motd is "Ansible Managed node by (your user name)"
    - Create a user with a variable defined in config.yaml

5. PUSH and COMMIT your PrelimExam in your GitHub repo

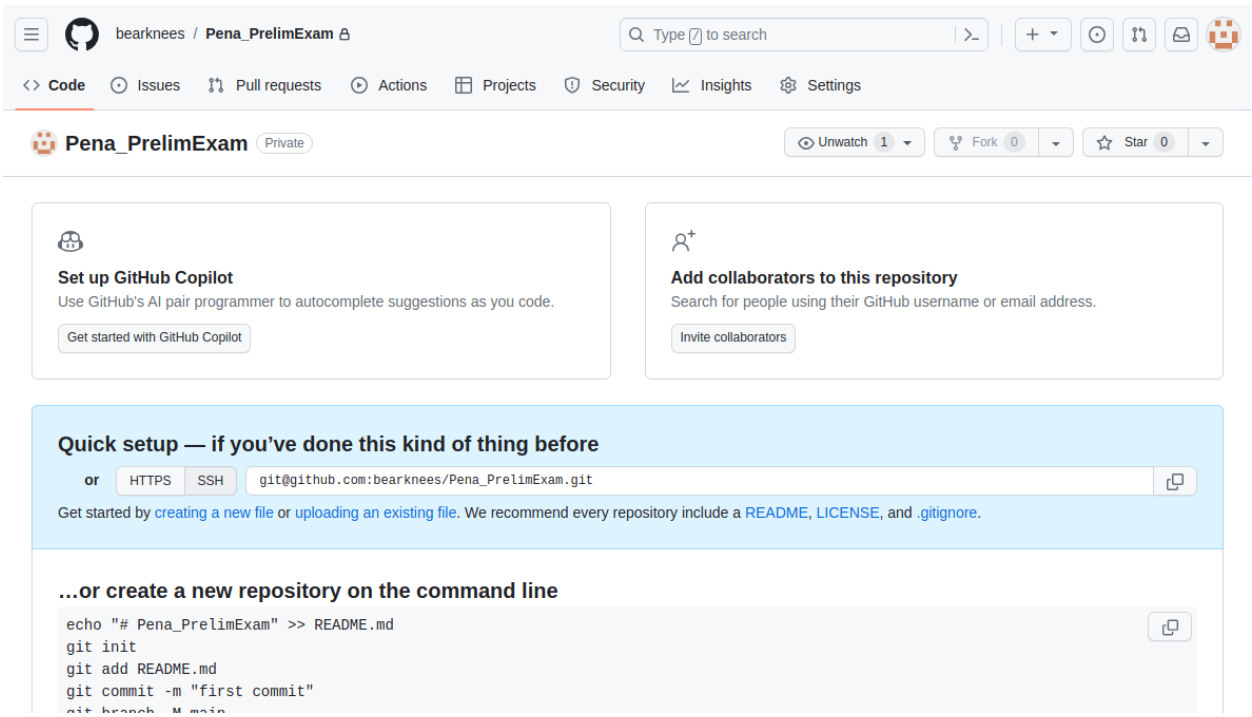
6. Your document report should be submitted here.

7. For your prelim exam to be counted, please paste your repository link here.

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

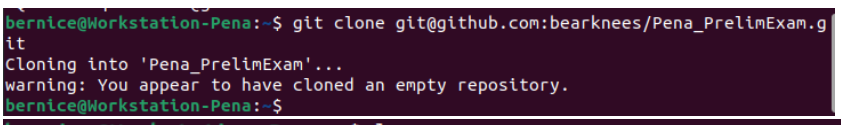


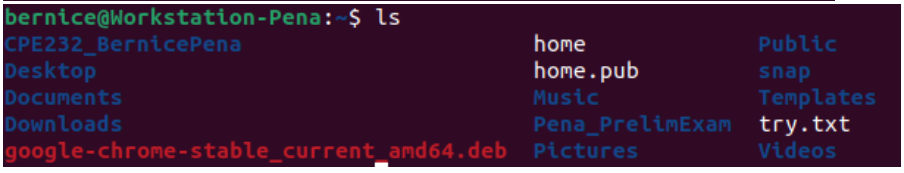
The screenshot shown above is where I was creating my new repository in github named Pena_PrelimExam.



Repository created

2. Clone your new repository in your CN.





After creating the repository in my GitHub account, I used the *git clone* command followed by the ssh link of my repository to clone it.

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

```
bernice@Workstation-Pena:~/Pena_PrelimExam$ nano inventory.ini
bernice@Workstation-Pena:~/Pena_PrelimExam$ nano ansible.cfg
```

I used the nano command to create the files.

```
p0s3
       valid_lft 84236sec preferred_lft 84236sec
    inet6 2001:4451:8519:8d00:23be:e522:1bea:4b7/64 scope global noprefixroute d
ynamic
       valid_lft 259128sec preferred_lft 172728sec
    inet6 fe80::20a0:7d70:1d0d:a9f4/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 08:00:27:47:01:f3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.106/24 brd 192.168.56.255 scope global noprefixroute dynamic
 enp0s8
       valid_lft 573sec preferred_lft 573sec
    inet6 fe80::1676:1532:c177:76d5/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 10
0
    link/ether 08:00:27:89:74:6a brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.18/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
       valid_lft 84243sec preferred_lft 84243sec
    inet6 2001:4451:8519:8d00:a00:27ff:fe89:746a/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 259166sec preferred_lft 172766sec
    inet6 fe80::a00:27ff:fe89:746a/64 scope link
       valid_lft forever preferred_lft forever
bernice@server1-pena:~$ _
```

Screenshots shown above is the ip address of my Ubuntu manage node and CentOS manage node.

```
bernice@Workstation-Pena: ~/Pena_PrelimExam
  GNU nano 6.2                    inventory.ini *
[ubuntu]
192.168.1.18

[centos]
192.168.56.106
```

This is what I have in my inventory file. I added the ip address of the Ubuntu manage node and CentOS manage node in this file since this is essential because it tells the Ansible which machines to target when developing automation tasks.

```
bernice@Workstation-Pena: ~/Pena_PrelimExam
  GNU nano 6.2                     ansible.cfg
[defaults]
inventory = inventory.ini
```

This is what I have in my ansible.cfg, I indicated there the inventory file I created earlier, the purpose of that is for the Ansible to tell which inventory file defines the hosts I want to configure or to manage since my inventory file consists of the ip addresses of my manage nodes.

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

- Installs the latest python3 and pip3
- use pip3 as default pip
- use python3 as default python
- Install Java open-jdk
- 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

```
bernice@Workstation-Pena:~/Pena_PrelimExam$ nano config.yaml
```

I used the nano command again to create the playbook under the repository Pena_PrelimExam.

bernice@Workstation-Pena: ~/Pena_PrelimExam

GNU nano 6.2 config.yaml

```
---
- hosts: all
  become: true
  vars:
    uname: "bernice"

  tasks:
    - name: Install Python3 in Ubuntu
      apt:
        name: python3
        state: latest
      when: ansible_distribution == "Ubuntu"

    - name: Install Python3 in CentOS
      dnf:
        name: python3
        state: latest
      when: ansible_distribution == "CentOS"

    - name: Install Pip3 in Ubuntu
      apt:
        name: python3-pip
        state: latest
      when: ansible_distribution == "Ubuntu"

    - name: Install Pip3 in CentOS
      dnf:
        name: python3-pip
        state: latest
      when: ansible_distribution == "CentOS"

    - name: Install Java in Ubuntu
      apt:
        name: default-jre
        state: latest
      when: ansible_distribution == "Ubuntu"
    - name: Install Java in CentOS
      dnf:
        name: java-11-openjdk-devel
        state: latest
      when: ansible_distribution == "CentOS"

    - name: Set Aliases for Python3 and Pip3
      shell: |
        echo 'alias python=/usr/bin/python3' >> ~/.bashrc
        echo 'alias pip=/home/{{ bernice }}/.local/bin/pip3' >> ~/.bashrc

    - name: Set MOTD
      debug:
        msg: "Ansible Managed node by {{ bernice }}"

    - name: Create User with Variable from config.yaml
      ansible.builtin.user:
        name: "{{ bernice }}"
        createhome: yes
        home: "/home/{{ bernice }}"
```

This is what's inside my config.yaml, the code shown above is to do the tasks such as

installing the latest python3 and pip3 using the pip3 as default pip (I also used python3 as default python), installing java open-jdk, create motd with the text defined by the default motd which is the Ansible Managed node by {{ bernice }}.

```
bernice@Workstation-Pena:~/Pena_PrelimExam$ ansible-playbook -i inventory.ini config.yaml --ask-become-pass
BECOME password:

PLAY [all] *********************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************
ok: [192.168.1.18]
ok: [192.168.56.106]

TASK [Install Python3 in Ubuntu] ***********************************************************************************************
skipping: [192.168.56.106]
ok: [192.168.1.18]

TASK [Install Python3 in CentOS] ***********************************************************************************************
skipping: [192.168.1.18]
ok: [192.168.56.106]

TASK [Install Pip3 in Ubuntu] **************************************************************************************************
skipping: [192.168.56.106]
ok: [192.168.1.18]

TASK [Install Pip3 in CentOS] **************************************************************************************************
skipping: [192.168.1.18]
ok: [192.168.56.106]

TASK [Install Java in Ubuntu] **************************************************************************************************
skipping: [192.168.56.106]
ok: [192.168.1.18]

TASK [Install Java in CentOS] **************************************************************************************************
skipping: [192.168.1.18]
ok: [192.168.56.106]

TASK [Set Aliases for Python3 and Pip3 (Ubuntu)] *******************************************************************************
skipping: [192.168.56.106]
ok: [192.168.1.18]

TASK [Set Aliases for Pip3 (Ubuntu)] *******************************************************************************************
skipping: [192.168.56.106]
ok: [192.168.1.18]

TASK [Set MOTD] ****************************************************************************************************************
skipping: [192.168.56.106]
ok: [192.168.1.18]

TASK [Create User with Variable from config.yaml] ******************************************************************************
ok: [192.168.1.18]
ok: [192.168.56.106]

PLAY RECAP *********************************************************************************************************************
192.168.1.18               : ok=8    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.106             : ok=5    changed=0    unreachable=0    failed=0    skipped=6    rescued=0    ignored=0

bernice@Workstation-Pena:~/Pena_PrelimExam$
```

This was the result when I executed the playbook using the ansible-playbook command, as you can see in the output, there are skips happened, this is because the tasks are being processed one by one, when it is happpening on the ubuntu manage node's ip, then the centos manage node's ip will be skipped and vice versa. Then it proceeded to install Python3, Pip3, and Java on the respective hosts, set up aliases for Python and Pip, and configured a Message of the Day (MOTD) for the Ubuntu host. Aside from this, it created a user with the username "bernice" as specified in the playbook's variables, then the play recap confirms that all tasks were executed without errors or issues on both hosts, demonstrating effective management and configuration of the target systems using Ansible.

Verification of the tasks in Ubuntu manage node:

```
bernice@server1-pena:~$ dpkg -l | grep python3
ii  libpython3-dev:amd64                 3.10.6-1~22.04                  amd64        header files and a static library for
 Python (default)
ii  libpython3-stdlib:amd64              3.10.6-1~22.04                  amd64        interactive high-level object-oriente
d language (default python3 version)
ii  libpython3.10:amd64                  3.10.12-1~22.04.2               amd64        Shared Python runtime library (versio
n 3.10)
ii  libpython3.10-dev:amd64              3.10.12-1~22.04.2               amd64        Header files and a static library for
 Python (v3.10)
```

Checking if Python 3 is installed

```
bernice@server1-pena:~$ java -version
openjdk version "11.0.20.1" 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
```

Checking if Java is installed

Verification of the tasks in CentOS manage node:

```
[bernice@pena_Workstation ~]$ rpm -q python3
python3-3.6.8-19.el7_9.x86_64
```

Checking if Python 3 is installed

```
[bernice@pena_Workstation ~]$ java -version
openjdk version "1.8.0_382"
OpenJDK Runtime Environment (build 1.8.0_382-b05)
OpenJDK 64-Bit Server VM (build 25.382-b05, mixed mode)
[bernice@pena_Workstation ~]$
```

Checking if Java is installed

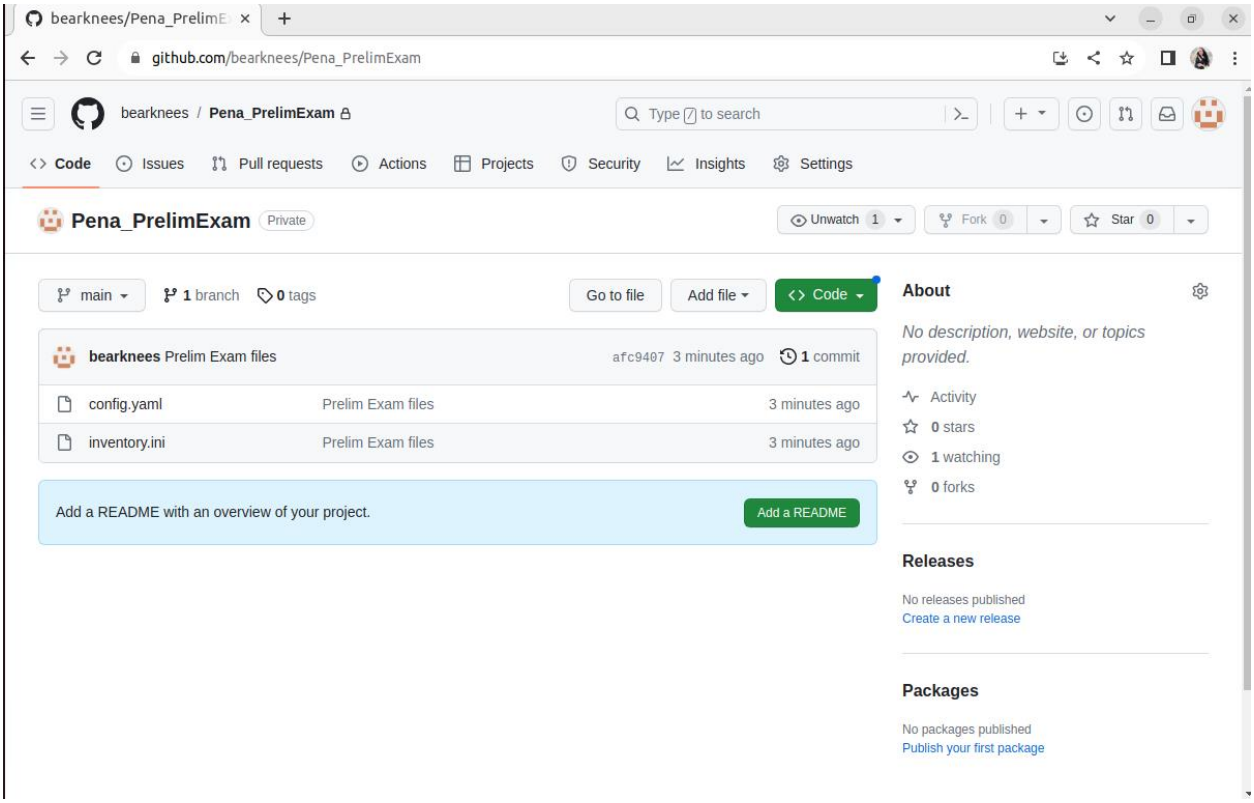
5. PUSH and COMMIT your PrelimExam in your GitHub repo

```
bernice@Workstation-Pena:~/Pena_PrelimExam$ git add config.yaml inventory.ini
bernice@Workstation-Pena:~/Pena_PrelimExam$ git commit -m "Prelim Exam files"
[main (root-commit) afc9407] Prelim Exam files
 2 files changed, 74 insertions(+)
 create mode 100644 config.yaml
 create mode 100644 inventory.ini
```

I added first the files I've created then I used the commit command to commit it under my repository.

```
bernice@Workstation-Pena:~/Pena_PrelimExam$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 752 bytes | 376.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:bearknees/Pena_PrelimExam.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
bernice@Workstation-Pena:~/Pena_PrelimExam$
```

I used the push command to push it in my GitHub repository.



I opened my GitHub account to confirm that the files have been pushed, and as you can see in the screenshot, the config.yaml and inventory is in my Pena_PrelimExam repository.

6. For your prelim exam to be counted, please paste your repository link here.

https://github.com/bearknees/Pena_PrelimExam