Name: Mark Andrei Ponayo
Course/Section: BSCPE31S5
Instructor: Engr. Roman Richard

Date Performed: 09/29/2023
Date Submitted: 10/1/2023
Semester and SY: 2nd 2023-2024

Hands-on Prelim Skill Exam

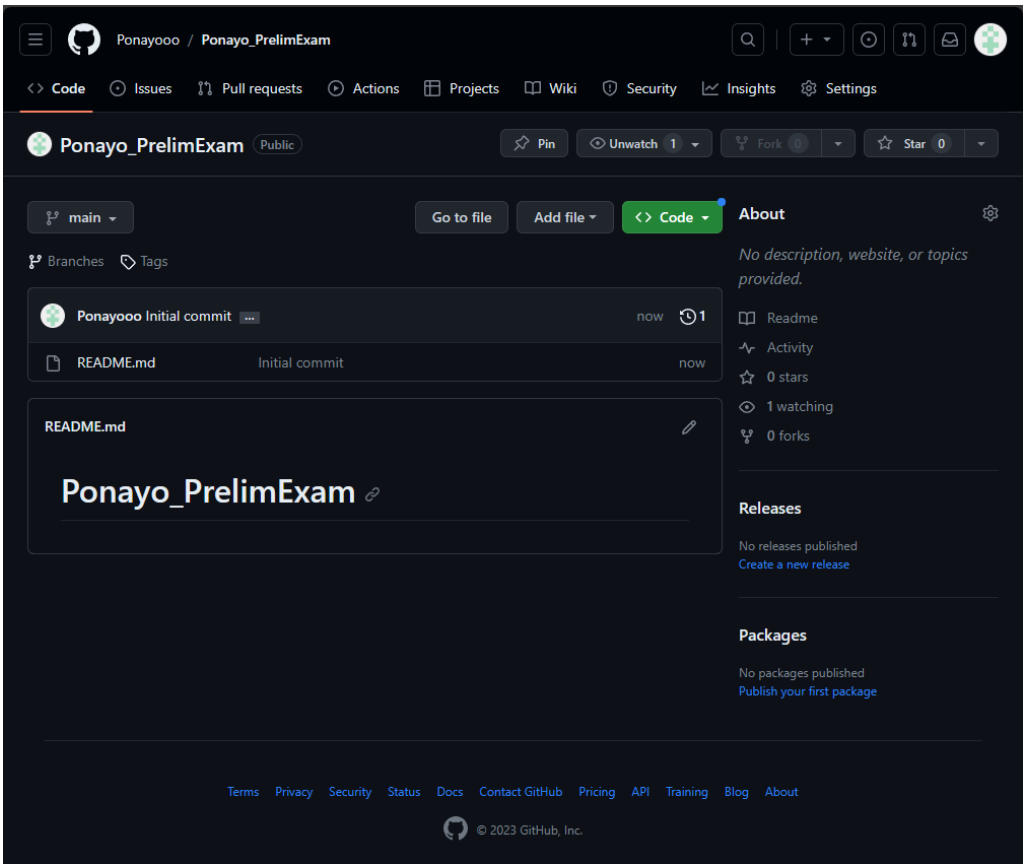
Tools Needed:

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

Procedure:

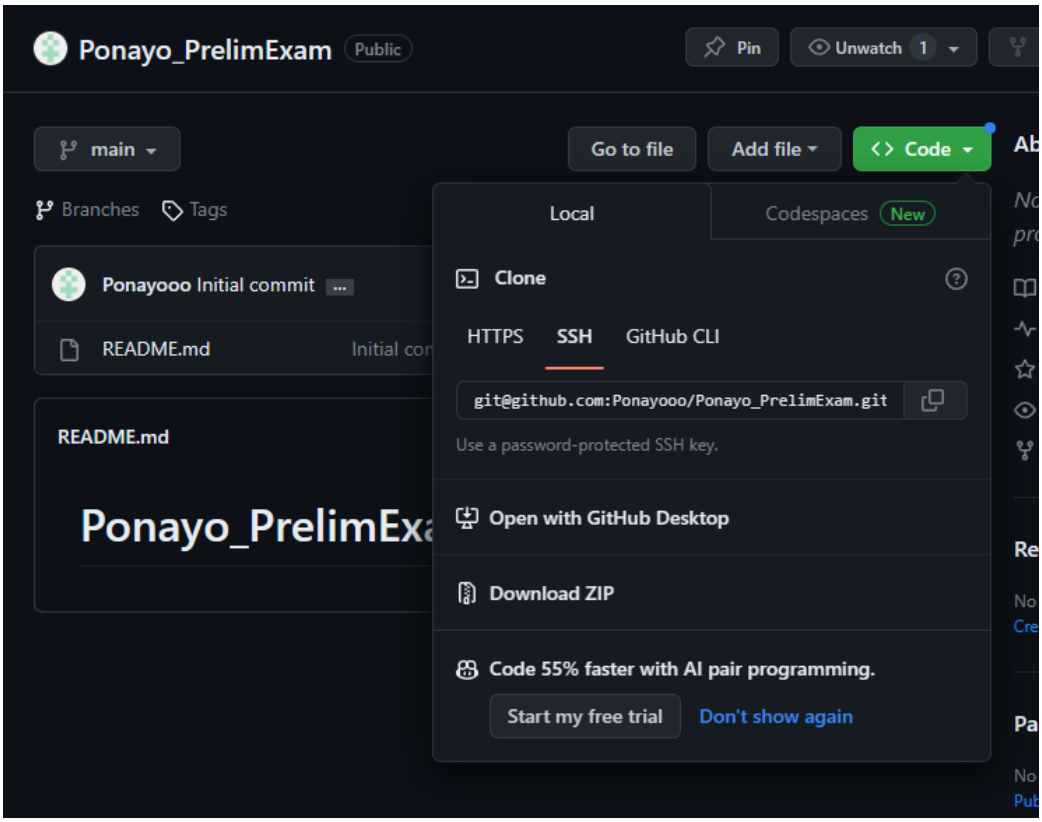
1. Note: You are required to create a document report of the steps you will do for this exam. All screenshots should be labeled and explained properly.

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



3. Clone your new repository in your CN.

Getting the ssh link before to clone



Using git clone command and the ssh link to connect the local and github.

```
ponayo@Workstation:~$ git clone git@github.com:Ponayooo/Ponayo_PrelimExam.git
Cloning into 'Ponayo_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
Receiving objects: 100% (3/3), done.
ponayo@Workstation:~$
```

On this screenshot, I used the ls command to show if the repository that i created is connected on the local host.

```
ponayo@Workstation:~$ ls
ansible.cfg    Desktop    Downloads  install_apache.yml  Pictures          Public  Templates
CPE232_Ponayo  Documents  hosts      Music               Ponayo_PrelimExam snap    Videos
ponayo@Workstation:~$
```

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

I used the cat command to show the value and created ansible.cfg and inventory.

```
ponayo@Workstation:~/Ponayo_PrelimExam$ cat ansible.cfg
[defaults]
inventory=hosts
host_key_checking=False

deprecation_warnings=False

remote_user=ponayo
private_key_file=~/.ssh/id_rsa
ponayo@Workstation:~/Ponayo_PrelimExam$ cat inventory
[servers]

192.168.56.110 ansible_python_interpreter=/usr/bin/python3
192.168.56.106 ansible_python_interpreter=/usr/bin/python3
192.168.56.107 ansible_python_interpreter=/usr/bin/python3
ponayo@Workstation:~/Ponayo_PrelimExam$
```

I use the ansible all -m ping to check the connections between the control node and manage node.

```
ponayo@Workstation:~/Ponayo_PrelimExam$ ansible all -m ping
192.168.56.106 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
192.168.56.107 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
192.168.56.110 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

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

Creating the script to install the python3 and pip 3 in Ubuntu and CentOS.

```
  GNU nano 6.2
---
- hosts: all
  become: true
  tasks:

  - name: install python3 and pip3 in Ubuntu
    apt:
      name:
        - python3
        - python3-pip
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install python3 in CentOS
    dnf:
      name:
        - python3
        - python3-pip
      state: latest
      use_backend: dnf4
      update_cache: yes
    when: ansible_distribution == "CentOS"
```

Executing the ansible-playbook to run and install the python3 in both localhost and CentOS

```
ponayo@Workstation:~/Ponayo_PrelimExam$ ansible-playbook --ask-become-pass config.yaml
BECOME password:

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

TASK [Gathering Facts] *****************************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.110]

TASK [install python3 and pip3 in Ubuntu] **********************************************************************
skipping: [192.168.56.110]
ok: [192.168.56.106]

TASK [install python3 in CentOS] *******************************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.110]

PLAY RECAP *****************************************************************************************************
192.168.56.106             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.110             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

- use pip3 as default pip
- use python3 as default python

The default script inside the inventory or manage node is set in python3

```
192.168.56.110 ansible_python_interpreter=/usr/bin/python3
192.168.56.110 ansible_connection=ssh

192.168.56.106 ansible_python_interpreter=/usr/bin/python3
192.168.56.106 ansible_connection=local
```

Setting up the inventory file as pip3 as a default.

```
[localhosts]

192.168.56.110 ansible_python_interpreter=/usr/bin/python3 pip_package=pip3
192.168.56.110 ansible_connection=ssh

192.168.56.106 ansible_python_interpreter=/usr/bin/python3 pip_package=pip3
192.168.56.106 ansible_connection=local
```

Then running the ansible-playbook to check if there are any errors.

```
ponayo@Workstation:~/Ponayo_PrelimExam$ ansible-playbook --ask-become-pass config.yaml
BECOME password:

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

TASK [Gathering Facts] *****************************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.110]

TASK [install python3 and pip3 in Ubuntu] **********************************************************************
skipping: [192.168.56.110]
ok: [192.168.56.106]

TASK [install python3 in CentOS] *******************************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.110]

PLAY RECAP *****************************************************************************************************
192.168.56.106             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.110             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

ponayo@Workstation:~/Ponayo_PrelimExam$
```

- Install Java open-jdk

Setting up the script for installing the java open-jdk for both Ubuntu and CentOS

```
- name: install java in Ubuntu
  apt:
    name: openjdk-11-jre
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: install java in CentOS
  dnf:
    name: java-11-openjdk
    state: latest
    update_cache: yes
    use_backend: dnf4
  when: ansible_distribution == "CentOS"
```

Run up the playbook and check if the script is applied or not.

```
ponayo@Workstation:~/Ponayo_PrelimExam$ ansible-playbook --ask-become-pass config.yaml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.110]

TASK [install python3 and pip3 in Ubuntu] **************************************************************************
skipping: [192.168.56.110]
ok: [192.168.56.106]

TASK [install python3 in CentOS] ***********************************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.110]

TASK [install java in Ubuntu] **************************************************************************************
skipping: [192.168.56.110]
changed: [192.168.56.106]

TASK [install java in CentOS] **************************************************************************************
skipping: [192.168.56.106]
changed: [192.168.56.110]

PLAY RECAP *********************************************************************************************************
192.168.56.106             : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.110             : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

ponayo@Workstation:~/Ponayo_PrelimExam$
```

As the recap on this screenshot. It shows that I successfully installed the java open-djk in both Ubuntu and CentOS.

- 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)"

Creating a script to create a motd.

```
- name: Motd message
  lineinfile:
    path: /etc/motd
    line: "{{ Hi! Welcome to Ansible Manage Node by Ponayo | default ('Ansible Managed Node by ' + ansible_user) }}"
  register: motd_result
  when: motd_message is defined

- name: Set Default Motd message
  lineinfile:
    path: /etc/motd
    line: "Ansible  Manage Node by {{ ansible_user }}"
  when: motd_message is not defined

- name: Display Motd result
  debug:
    msg: "{{ lookup('file','/etc/motd') }}"
```

By running the ansible-playbook this is the output of the script that i created.

```
TASK [Set Default Motd message] ************************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.110]

TASK [Display Motd result] *****************************************************************************************
ok: [192.168.56.110] => {
    "msg": "Ansible  Manage Node by ponayo"
}
ok: [192.168.56.106] => {
    "msg": "Ansible  Manage Node by ponayo"
}

PLAY RECAP *********************************************************************************************************
192.168.56.106             : ok=5    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.110             : ok=5    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

On this screenshot, it shows that I successfully created a motd that shows the default motd “Ansible Manage Node by Ponayo if there are no input variables.

- Create a user with a variable defined in config.yaml

Creating a script in adding a user to which a variable is defined.

```
- name: Create a new user
  user:
    name: "{{ new_username }}"
    state: present
  when: new_username is defined
```

This is the output after when i run the ansible-playbook.

```
TASK [Create a new user] ***************************************************************************************
skipping: [192.168.56.110]
skipping: [192.168.56.106]

PLAY RECAP *****************************************************************************************************
192.168.56.106             : ok=5    changed=0    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
192.168.56.110             : ok=5    changed=0    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
```

5. PUSH and COMMIT your PrelimExam in your GitHub repo

Checking up the status of the files in my repository.

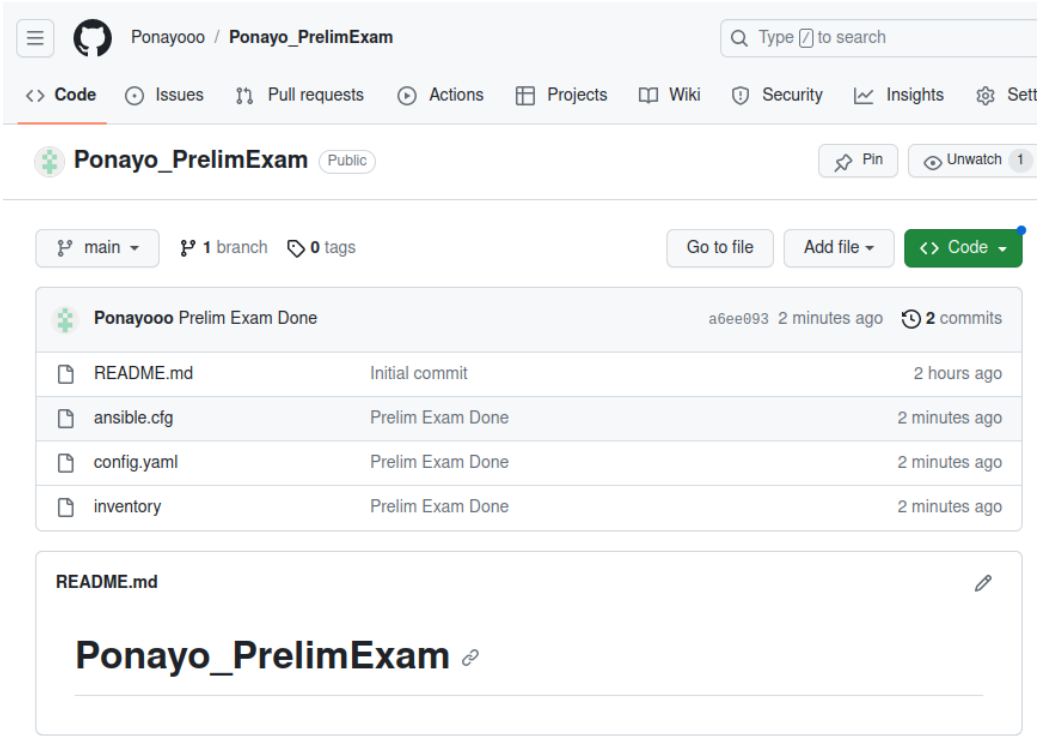
```
ponayo@Workstation:~/Ponayo_PrelimExam$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ansible.cfg
        config.yaml
        inventory
```

Adding all the files that i used to my git and inserting a commit message.

```
ponayo@Workstation:~/Ponayo_PrelimExam$ git commit -m "Prelim Exam Done"
[main a6ee093] Prelim Exam Done
 3 files changed, 76 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 config.yaml
 create mode 100644 inventory
```

This screenshot shows that i successfully push my files in github.



6. Your document report should be submitted here.

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

Ponayooo/Ponayo_PrelimExam (github.com)