Name: Rustom C. Cariño	**Date Performed:9/29/2023**
Course/Section:CPE31S5	**Date Submitted:9/30/2023**
Instructor: Engr. Roman Richard	**Semester and SY: 1st semester/ 2023-2024**

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**

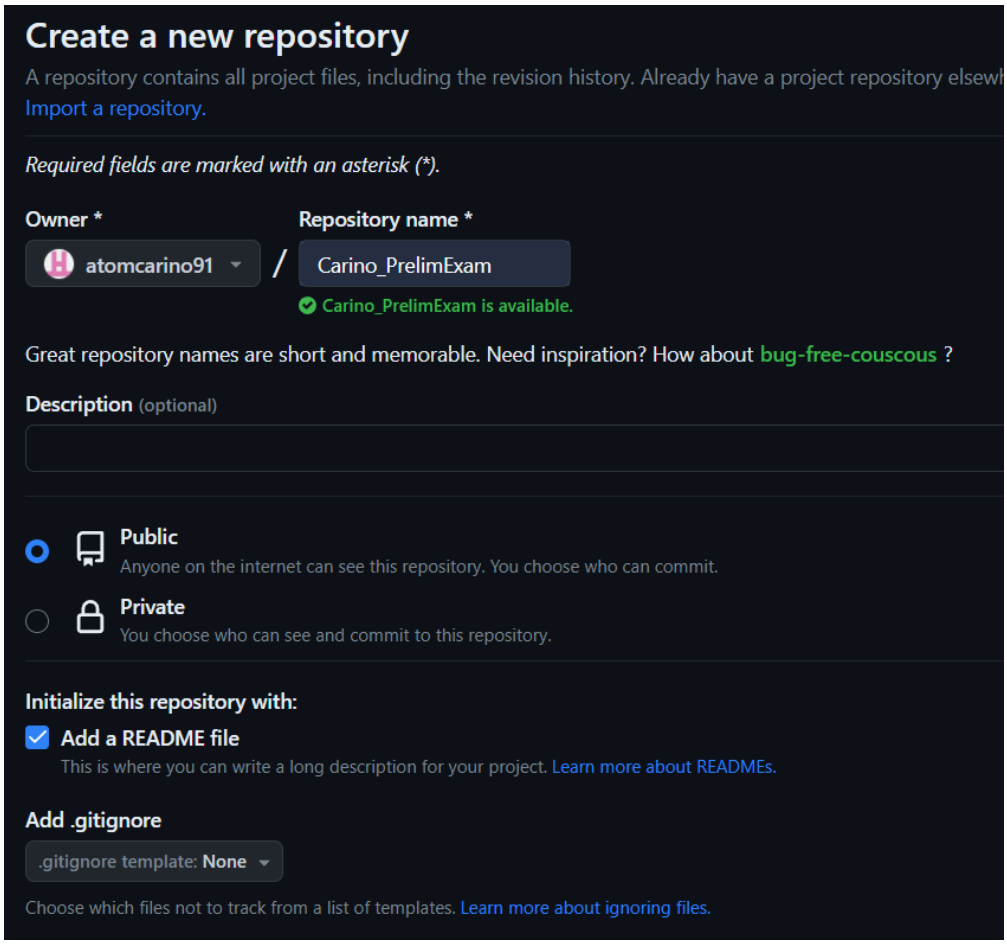

Figure 2.1 Creating a new repository in Github

- In Github I created a new public repository named Carino_Prelim exam.

3. **Clone your new repository in your CN.**

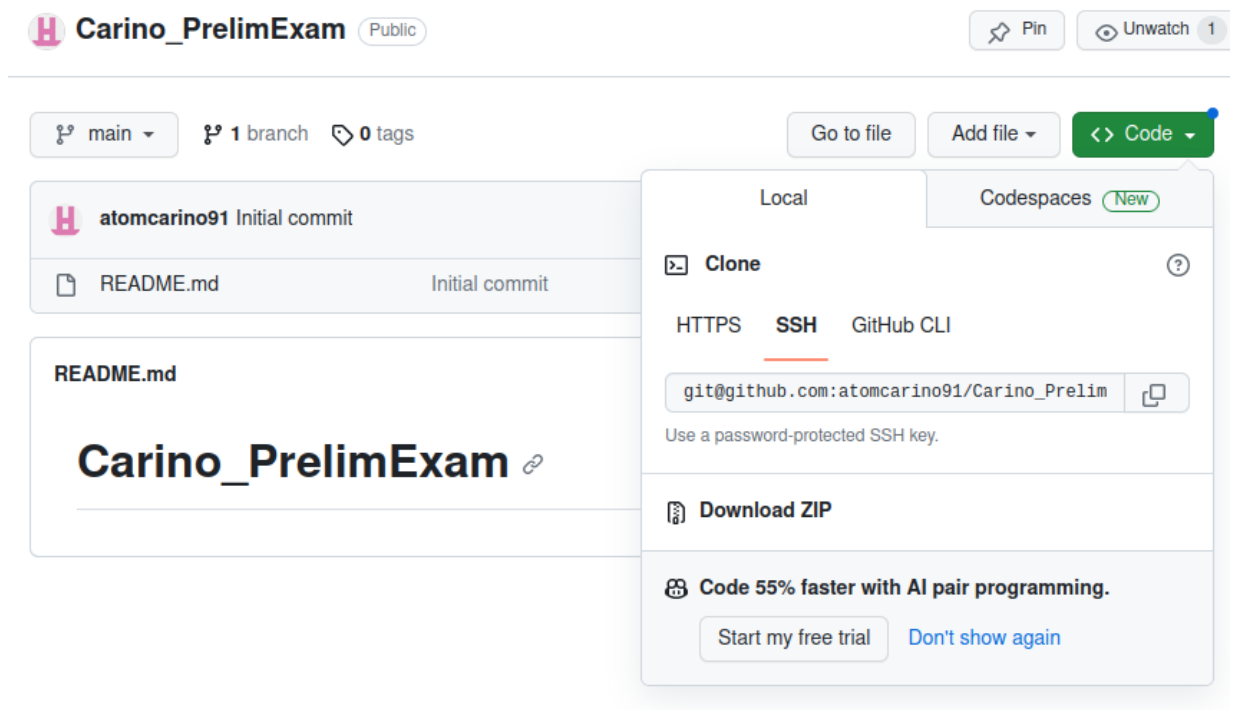

Figure 3.1 Copying the ssh link of repository

- Copying ssh link of the repository to be pasted in the terminal of the localmachine.

```
rustom@localmachine:~$ git clone git@github.com:atomcarino91/Carino_PrelimExam.git
Cloning into 'Carino_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.
rustom@localmachine:~$
```

Figure 3.2 Cloning the repository

- Using the git clone command and pasting the ssh link will clone the repository from the github account to my control node.

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

```
rustom@localmachine:~/Carino_PrelimExam$ sudo nano inventory
[sudo] password for rustom:
```

Figure 4.1.1 Creating inventory file

- For creating an inventory file, I used sudo nano “inventory”.

```
rustom@localmachine: ~/Carino_PrelimExam
  GNU nano 6.2                          inventory *
#Server1
192.168.56.108 ansible_python_interpreter=/usr/bin/python3
192.168.56.108 apache_package=apache2 php_package=libapache2-mod-php

#CentOS
192.168.56.118 ansible_python_interpreter=/usr/bin/python3
192.168.56.118 apache_package=httpd php_package=php
```

Figure 4.1.2 Content of inventory

- For the content of the inventory, I just copy the content of the previous activity since it has the same managed nodes.

```
rustom@localmachine:~/Carino_PrelimExam$ sudo nano ansible.cfg
rustom@localmachine:~/Carino_PrelimExam$
```

Figure 4.2.1 Creating ansible.cfg

- For creating the ansible.cfg, it was the same as when I created the inventory file.

```
rustom@localmachine: ~/Carino_PrelimExam
  GNU nano 6.2                          ansible.cfg *
[defaults]

inventory=hosts
host_key_checking=False

deprecation_warning=False

remote_user=rustom
private_key_file=~/.ssh/id_rsa
```

Figure 4.2.2 Content of ansible.cfg

- The content of the ansible.cfg is also the same as my previous activity. As you can see the inventory is set to the hosts variable because I'm having trouble if the inventory was set to the inventory itself.

```
rustom@localmachine: ~/Carino_PrelimExam
GNU nano 6.2                                hosts
[localhosts]

192.168.56.108 ansible_connection=local
192.168.56.118 ansible_connection=ssh
```

Figure 4.3.1 Content of hosts

- The hosts is just an additional file. This file is the file that variable was used to specify the command that ansible will be able to use to connect on the remote hosts. As you can see in my ansible.cfg, the inventory is set to the hosts file.

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

```
rustom@localmachine:~/Carino_PrelimExam$ sudo nano config.yaml
rustom@localmachine:~/Carino_PrelimExam$
```

Figure 5.1.1 Creating a config.yam

- I created the config.yaml using the sudo nano command.
- **Installs the latest python3 and pip3**

```
rustom@localmachine: ~/Carino_PrelimExam
GNU nano 6.2                                config.yaml
---
#Carino PrelimExam

- hosts: all
  become: true

  tasks:
  - name: Install latest python3 and pip3
    package:
      name:
      - python3
      - python3-pip
      state: latest
      update_cache: yes
```

Figure 5.2.1 Installs the latest python3 and pip3 using playbook

- I used the package instead of apt or dnf because it is more efficient to use the package since when I use the apt or dnf it will need two different tasks but the same function.

```
rustom@localmachine:~/Carino_PrelimExam$ ansible-playbook --ask-become-pass config.yaml
BECOME password:

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

TASK [Gathering Facts] ***********************************************************
**********
ok: [192.168.56.108]
ok: [192.168.56.118]

TASK [Install latest python3 and pip3] *******************************************
**********
ok: [192.168.56.108]
ok: [192.168.56.118]

PLAY RECAP ***********************************************************************
**********
192.168.56.108             : ok=2    changed=0    unreachable=0    failed=0    skipped=0
   rescued=0    ignored=0
192.168.56.118             : ok=2    changed=0    unreachable=0    failed=0    skipped=0
   rescued=0    ignored=0
```

Figure 5.2.2 Executing the config.yaml

- Both managed node was successfully installed the python3 and pip3

- **use pip3 as default pip**

```
rustom@localmachine: ~/Carino_PrelimExam
  GNU nano 6.2                                   inventory
[servers]

#Server1
192.168.56.108 ansible_python_interpreter=/usr/bin/python3 pip_package=pip3

#CentOS
192.168.56.118 ansible_python_interpreter=/usr/bin/python3 pip_package=pip3
```

Figure 5.3.1 Using the pip3 as default pip

- In the inventory file, I used pip3 as default pip in both managed nodes.

- **use python3 as default python**

```
rustom@localmachine: ~/Carino_PrelimExam
  GNU nano 6.2                                   inventory
[servers]

#Server1
192.168.56.108 ansible_python_interpreter=/usr/bin/python3 pip_package=pip3

#CentOS
192.168.56.118 ansible_python_interpreter=/usr/bin/python3 pip_package=pip3
```

Figure 5.4.1 Using python3 as default python

- I just combined the pip3 as default pip and python3 as default python in one line for efficiency.

- **Install Java open-jdk**

```yaml
- name: Install java open-jdk in ubuntu
  apt:
    name:
    - openjdk-17-jdk
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: Install java open-jdk in centos
  dnf:
    name:
      - java-11-openjdk
    state: latest
    update_cache: yes
  when: ansible_distribution == "Centos"
```

Figure 5.5.1 Installing Java open-jdk in both Manage Nodes

- In this case I will not be able to use the package command since there was a specific syntax for ubuntu server and centos, therefore I created two modules inside the one task for ubuntu and centos.

```
rustom@localmachine:~/Carino_PrelimExam$ ansible-playbook --ask-become-pass config.yaml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.108]
ok: [192.168.56.118]

TASK [Install latest python3 and pip3] *****************************************************************************
ok: [192.168.56.108]
ok: [192.168.56.118]

TASK [Install java open-jdk in ubuntu] *****************************************************************************
skipping: [192.168.56.118]
ok: [192.168.56.108]

TASK [Install java open-jdk in centos] *****************************************************************************
skipping: [192.168.56.108]
skipping: [192.168.56.118]

PLAY RECAP *********************************************************************************************************
192.168.56.108             : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.118             : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Figure 5.5.2 Executing the config.yaml

- Both centos and ubuntu was successfully installed the 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)"**

```
vars:
  motd:
    - Ansible Managed node by rustom
```

Figure 5.6.1 Message of the day variable

- The vars syntax is used to define a variable which is in the motd, and the motd holds the text “Ansible Managed node by rustom”.

```
tasks:
- name: Banner MOTD
  ansible.builtin.debug:
  msg:
  - "{{ motd }}"
```

Figure 5.6.2 Module of MOTD

- The ansible.builtin.debug is a module that debugs the statement without halting the playbook. The msg command is used to print a message which is the motd variable that holds the “Ansible Managed node by rustom”.

```
TASK [Banner MOTD] ******************************
ok: [192.168.56.108] => {
    "msg": [
        [
            "Ansible Managed Node by rustom"
        ]
    ]
}
ok: [192.168.56.118] => {
    "msg": [
        [
            "Ansible Managed Node by rustom"
        ]
    ]
}
```

Figure 5.6.3 Executing MOTD

- The MOTD banner task is successfully executed in both managed nodes and it displays the motd text since there is no variable inputted.

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

```
vars_prompt:
    - name: username
      prompt: Input username
      private: false

    - name: uid
      prompt: Input UID
      private: false
```

Figure 5.7.1 Variable that prompt the user

- The vars_prompt is used to prompt the user in inputting their username and uid. For the private command I used false to see what I just typed. If the private is true, you can't see what your typing and thats why is used false since it is difficult to type on the keyboard without seeing what I am typing.

```
- name: Create a User
  ansible.builtin.user:
    name: "{{ username }}}"
    comment: New User
    uid: "{{ uid }}"
    createhome: yes
    home: /home/"{{ username }}"
    shell: /bin/bash
```

Figure 5.7.2 The module that create the user

- In this module is where the user is created and where it will be stored.

6. PUSH and COMMIT your PrelimExam in your GitHub repo

```
rustom@localmachine:~/Carino_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

nothing added to commit but untracked files present (use "git add" to track)
rustom@localmachine:~/Carino_PrelimExam$ git add *
rustom@localmachine:~/Carino_PrelimExam$ git commit -m "Prelim Exam"
[main c21fc93] Prelim Exam
 3 files changed, 72 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 config.yaml
 create mode 100644 inventory
rustom@localmachine:~/Carino_PrelimExam$ git push origin
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 989 bytes | 989.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:atomcarino91/Carino_PrelimExam.git
   1a5798d..c21fc93  main -> main
rustom@localmachine:~/Carino_PrelimExam$
```

Figure 6.1 Push and commit to GitHub Repository

- First I check what I am pushing to my repository using the git status command. Then committing the files with a message “Prelim Exam” then pushing the 3 files to my github repository.

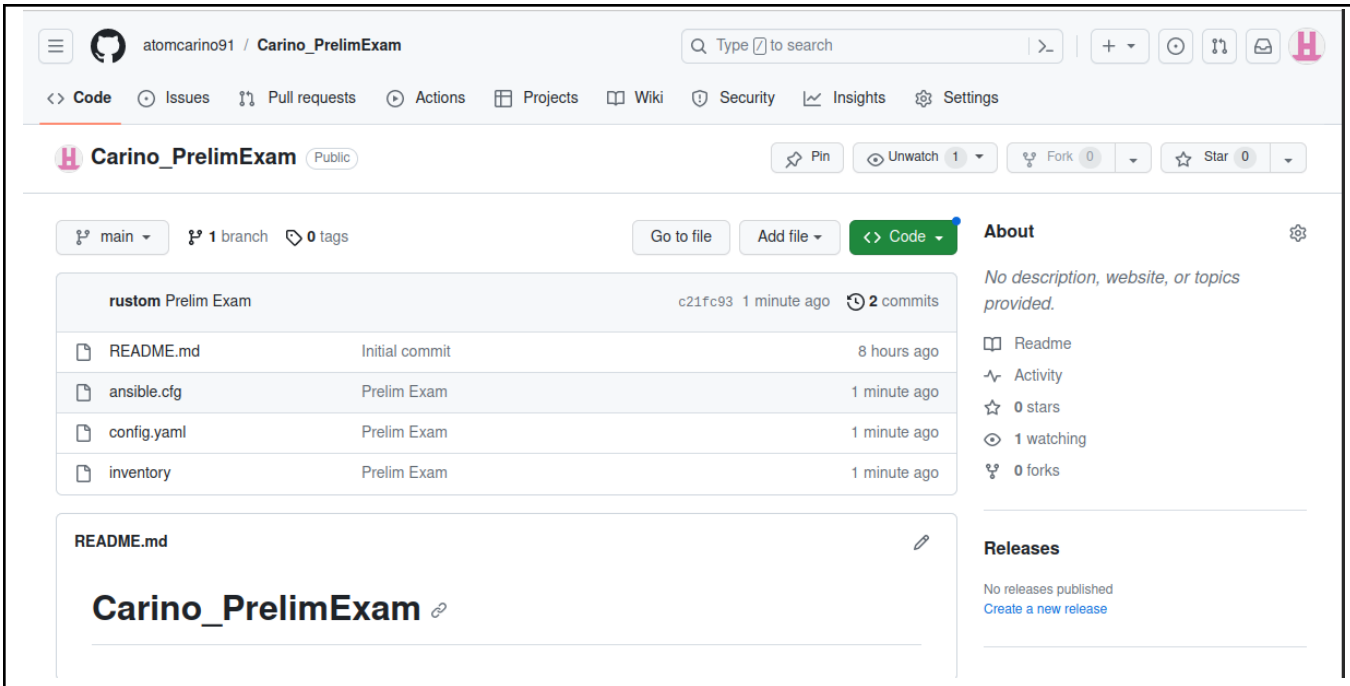

Figure 6.2 Checking if the pushing and committing is successful

- The three files that I push and commit are already updated to my repository in GitHub.

7. Your document report should be submitted here.

```yaml
---
#Carino PrelimExam
- hosts: all
  become: true

  vars:
    motd:
      - Ansible Managed Node by rustom

  vars_prompt:
    - name: username
      prompt: Input your username
      private: false

    - name: uid
      prompt: Input your UID
      private: false

  tasks:
  - name: Banner MOTD
    ansible.builtin.debug:
      msg:
        - "{{ motd }}"

  - name: Install latest python3 and pip3
    package:
      name:
      - python3
      - python3-pip
      state: latest
      update_cache: yes

  - name: Install java open-jdk in ubuntu
    apt:
      name:
      - openjdk-17-jdk
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: Install java open-jdk in centos
    dnf:
      name:
        - java-11-openjdk
      state: latest
      update_cache: yes
    when: ansible_distribution == "Centos"

  - name: Create a user
    ansible.builtin.user:
      name: "{{ username }}"
      comment: New User
      uid: "{{ uid }}"
      createhome: yes
      home: /home/"{{ username }}"
      shell: /bin/bash
```

Figure 7.1 Entire config.yaml file

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

https://github.com/atomcarino91/Carino_PrelimExam.git