

Name: Renier L. Lope	Date Created: 09/18/2024
Course & Section: CPE212 – CPE31S2	Date Submitted: 09/18/2024
Instructor: Engr. Robin Valenzuela	Sem&SchoolYear: 1st Sem (2024–2025)

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. LABELED AND EXPLAIN EACH CODE (PLAYBOOK) No explanation = Minus Points
- 2. Create a repository in your GitHub account and label it as Surname_PrelimExam

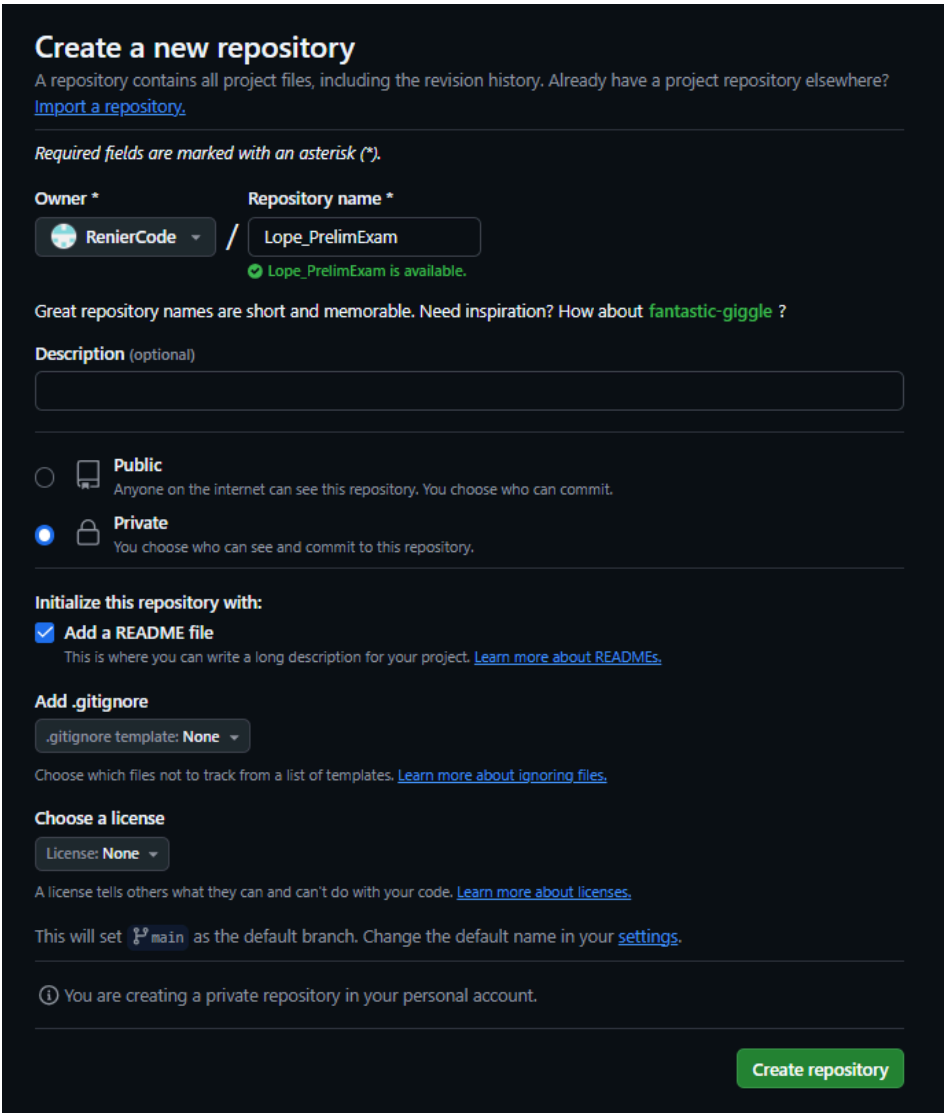


Figure 2.1: Creating a New Repository named (Lope_PrelimExam)

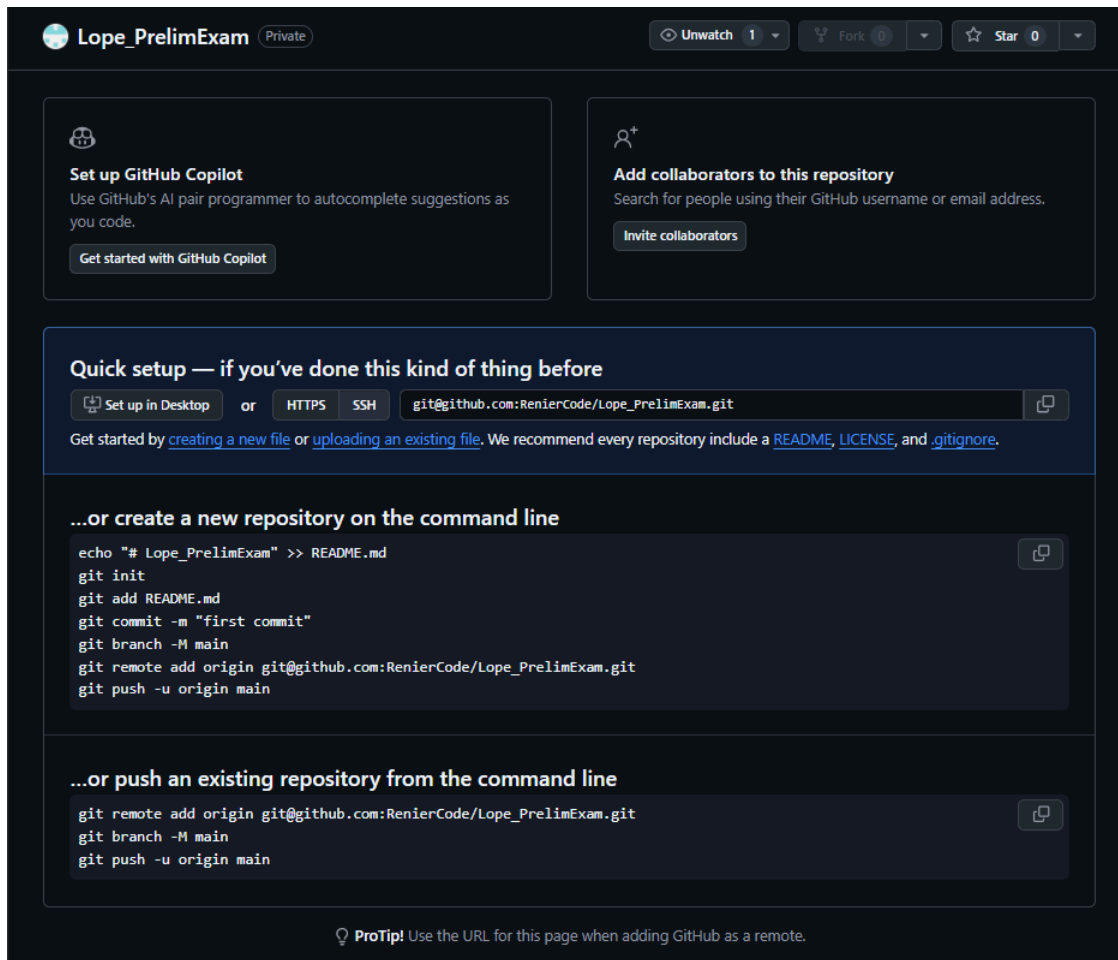


Figure 2.2: The Created Repository (Lope_PrelimExam)

3. Clone your new repository in your CN.

```
rnrlope@workstation:~$ git clone git@github.com:RenierCode/Lope_PrelimExam.git
Cloning into 'Lope_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.
rnrlope@workstation:~$
```

Figure 3.1: Cloning the repository to local machine using “git clone ssh-link-of-repository”

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

```
rnrlope@workstation:~/Lope_PrelimExam$ nano inventory
rnrlope@workstation:~/Lope_PrelimExam$ nano ansible.cfg
rnrlope@workstation:~/Lope_PrelimExam$
```

Figure 4.1: Creating an inventory and ansible.cfg files using nano.

```

rnrlope@workstation: ~/Lope_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 inventory
[ManageNodes]
server1
server2
```

Figure 4.2: Inputting the hostnames for the Manage Nodes

```

rnrlope@workstation: ~/Lope_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 ansible.cfg
[defaults]
inventory = inventory
remote_user = rnrlope
host_key_checking = True
```

Figure 4.3: Inputting the defaults of ansible.cfg

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

```
rnrlope@workstation:~/Lope_PrelimExam$ nano config.yaml
```

Figure 5.1: Creating an ansible playbook file named 'config.yaml' file using nano

- Installs the latest python3 and pip3

```

rnrlope@workstation: ~/Lope_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 config.yaml
---
- hosts: all
  become: true
  tasks:

  - name: install the latest python3 and pip3
    apt:
      name: python3-pip
      state: latest
```

Figure 5.2.1: Creating a task inside the playbook that will install python3 and pip3 inside the Manage Nodes

```

TASK [installs the latest python3 and pip3] *****
*
changed: [server2]
changed: [server1]

PLAY RECAP *****
*
server1           : ok=2    changed=1    unreachable=0    failed=0
server2           : ok=2    changed=1    unreachable=0    failed=0

```

Figure 5.2.2: Output of running the playbook

```

rnrlope@server1:~$ python3 --version
Python 3.6.9
rnrlope@server1:~$ pip3 --version
pip 9.0.1 from /usr/lib/python3/dist-packages (python 3.6)
rnrlope@server1:~$

```

Figure 5.2.3: Verifying the installation in server1

```

rnrlope@server2:~$ pip3 --version
pip 9.0.1 from /usr/lib/python3/dist-packages (python 3.6)
rnrlope@server2:~$ python3 --version
Python 3.6.9
rnrlope@server2:~$

```

Figure 5.2.4: Verifying the installation in server2

- use pip3 as default pip

```

[ManageNodes:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_pip_interpreter=/usr/bin/pip3

```

Figure 5.3.1: Creating variables inside the inventory to specify the version of pip

```

rnrlope@workstation: ~/Lope_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 config.yml
---
- hosts: all
  become: true
  tasks:
    - name: install the latest python2 and pip3
      apt:
        name: python3-pip
        state: latest
      vars:
        ansible_pip_interpreter: /usr/bin/pip3

```

Figure 5.3.2: Calling the variable to pip3 as default pip

- use python3 as default python

```
[ManageNodes:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_pip_interpreter=/usr/bin/pip3
```

Figure 5.4.1: Creating variables inside the inventory to specify the version of pip

```
rnrllope@workstation: ~/Lope_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 config.yml
---
- hosts: all
  become: true
  tasks:
  - name: install the latest python2 and pip3
    apt:
      name: python3-pip
      state: latest
    vars:
      ansible_pip_interpreter: /usr/bin/pip3
      ansible_python_interpreter: /usr/bin/python3
```

Figure 5.4.2: Calling the variable to pip3 as default pip

- Install Java open-jdk

```
rnrllope@workstation: ~/Lope_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 config.yml
---
- hosts: all
  become: true
  tasks:
  - name: install the latest python2 and pip3
    apt:
      name: python3-pip
      state: latest
    vars:
      ansible_pip_interpreter: /usr/bin/pip3
      ansible_python_interpreter: /usr/bin/python3
  - name: install Java open-jdk
    apt:
      name: openjdk-17-jre
      state: latest
```

Figure 5.5.1: Creating a task inside the playbook that will install Java openjdk inside the Manage Nodes

```
TASK [install Java open-jdk] *****
*
changed: [server1]
changed: [server2]

PLAY RECAP *****
*
server1      : ok=3    changed=1    unreachable=0    failed=0
server2      : ok=3    changed=1    unreachable=0    failed=0
```

Figure 5.5.2: Creating a task inside the playbook that will install Java openjdk inside the Manage Nodes

- 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
    state: latest
    update_cache: yes

- name: start and enable mariadb server
  service:
    name: mariadb
    enabled: true
    state: started
```

- 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

6. PUSH and COMMIT your PrelimExam in your GitHub repo

```
rnrllope@workstation:~/Lope_PrelimExam$ git add ansible.cfg
rnrllope@workstation:~/Lope_PrelimExam$ git add inventory
rnrllope@workstation:~/Lope_PrelimExam$ git add config.yml
rnrllope@workstation:~/Lope_PrelimExam$ git commit -m "Prelim"
```

```
rnrllope@workstation:~/Lope_PrelimExam$ git commit -m "Prelim"
[main 1808e25] Prelim
3 files changed, 41 insertions(+)
create mode 100644 ansible.cfg
create mode 100644 config.yml
create mode 100644 inventory
rnrllope@workstation:~/Lope_PrelimExam$ git push origin main
Counting objects: 5, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 742 bytes | 742.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To github.com:RenierCode/Lope_PrelimExam.git
5dcb5e4..1808e25 main -> main
rnrllope@workstation:~/Lope_PrelimExam$
```

7. Your document report should be submitted here.

8. For your prelim exam to be counted, please paste your repository link here. (Failure to submit will result in ZERO)

9. NO USE OF EXTERNAL WEBSITES SUCH AS , REDDIT, CHATGPT, GITHUB, GEMINI, CLAUDE, FORUMS, AND DOCUMENTATIONS. FAILURE TO COMPLY WITH RESULT IN ZERO.

GITHUB LINK: https://github.com/RenierCode/Lope_PrelimExam.git

Conclusion and Learnings:

- In this Prelim Skill Exam, I manage to create an ansible playbook that will install the latest versions of python3, pip3, Java openjdk, and MariaDB and will set pip3 and python3 as the default pip and python. I learned how to apply defaults and also to create a task on my own. Unfortunately I didn't manage to create a table.