| Hands-on Prelim Exam | |
|---|---|
| **Name:** Torrecampo, Juan Piolo S. | **Date Performed:** Sep 20, 2022 |
| **Course/Section:** CPE 232 / CPE31S22 | **Date Submitted:** Sep 20, 2022 |
| **Instructor:** Dr. Jonathan Taylar | **Semester and SY:** 1st Sem, 2022 - 2023 |

**Tools Needed**

- Control Node (CN) - 1
- Manage Node (MN) - 1 Ubuntu
- 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.

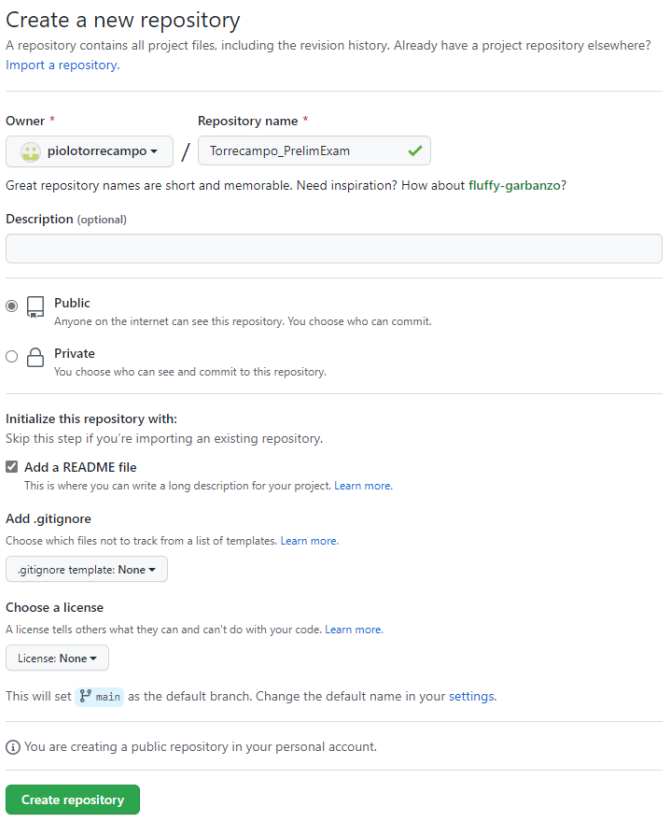


**Figure 1.1. The screenshot above shows the Github page where I am in the process of creating a**

**new repository named “Torrecampo_PrelimExam”.**

3. Clone your new repository in your CN.

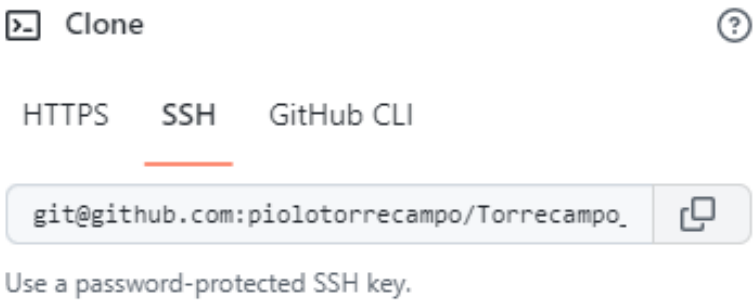


**Figure 1.2. Copying the ssh link of the created repository.**

```
piolo@workstation:~$ git clone git@github.com:piolotorrecampo/Torrecampo_PrelimE
xam
Cloning into 'Torrecampo_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.
piolo@workstation:~$ cd Torrecampo_PrelimExam
piolo@workstation:~/Torrecampo_PrelimExam$
```

**Figure 1.3 Cloning the new repository using “git clone ssh_link‘ command, where the ssh_link is the link for the github repository. After cloning, I change my current directory to the repository directory.**

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

```
piolo@workstation:~/Torrecampo_PrelimExam$ touch inventory ansible.cfg
piolo@workstation:~/Torrecampo_PrelimExam$ ll
total 16
drwxrwxr-x  3 piolo piolo 4096 Sep 20 07:53 ./
drwxr-x--- 19 piolo piolo 4096 Sep 20 07:50 ../
-rw-rw-r--  1 piolo piolo    0 Sep 20 07:53 ansible.cfg
drwxrwxr-x  8 piolo piolo 4096 Sep 20 07:50 .git/
-rw-rw-r--  1 piolo piolo    0 Sep 20 07:53 inventory
-rw-rw-r--  1 piolo piolo   23 Sep 20 07:50 README.md
piolo@workstation:~/Torrecampo_PrelimExam$
```

**Figure 1.4. Creating the two files using the “touch” command and verifying it using “ll” command.**

```
[ubuntu]
server1 ansible_connection=ssh ansible_user=piolo
[CentoOs]
centos_gui ansible_connection=ssh ansible_user=piolo
```

Figure 1.5. The screenshot above shows the contents of the inventory file.

```
[defaults]
inventory = inventory
host_key_checking = False
deprecation_warnings = False
private_key_file = ~/.ssh/id_rsa
```

Figure 1.6. The screenshot above shows the contents of the ansible.cfg file.

```
piolo@workstation:~/Torrecampo_PrelimExam$ ssh piolo@centos_gui
Last login: Mon Sep 19 20:00:44 2022
[piolo@localhost ~]$ exit
logout
Connection to centos_gui closed.
piolo@workstation:~/Torrecampo_PrelimExam$ ssh piolo@server1
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

25 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Tue Sep  6 10:42:49 2022 from 192.168.56.102
piolo@server1:~$ exit
logout
Connection to server1 closed.
piolo@workstation:~/Torrecampo_PrelimExam$
```

Figure 1.7. Checking the ssh connection between control nodes and manage nodes.

```
piolo@workstation:~/Torrecampo_PrelimExam$ ansible all -m ping
server1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
centos_gui | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
piolo@workstation:~/Torrecampo_PrelimExam$
```

**Figure 1.8. Testing pinging the two managed nodes using the "ansible all -m ping" set of commands.**

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

```
piolo@workstation:~/Torrecampo_PrelimExam$ touch config.yml
piolo@workstation:~/Torrecampo_PrelimExam$ ll
total 16
drwxrwxr-x  3 piolo piolo 4096 Sep 20 07:55 ./
drwxr-x--- 19 piolo piolo 4096 Sep 20 07:50 ../
-rw-rw-r--  1 piolo piolo    0 Sep 20 07:53 ansible.cfg
-rw-rw-r--  1 piolo piolo    0 Sep 20 07:55 config.yml
drwxrwxr-x  8 piolo piolo 4096 Sep 20 07:50 .git/
-rw-rw-r--  1 piolo piolo    0 Sep 20 07:53 inventory
-rw-rw-r--  1 piolo piolo   23 Sep 20 07:50 README.md
piolo@workstation:~/Torrecampo_PrelimExam$
```

**Figure 1.9. The screenshot above shows the creation of "config.yml" and verifying it using the "ll" command.**

```yaml
---
- hosts: all
  become: true
  vars:
    # Packages
    - python: python3
    - pip3: python3-pip
    # Creating a user
    - name_var: papzi
    - state_var: present
    - shell_var: /bin/bash
    - system_var: no
    - createhome_var: yes
    - home_var: /home/papzi
    # Motd
    - motd_var: "Amsible Managed node by Piolo"
  tasks:

  - name: Installing python3, pip3 and openjdk
    package:
      name:
        - "{{python}}"
        - "{{pip3}}"
        - "{{javaopenjdk}}"
      state: latest

  - name: Setting python3 interpreter as default
    shell: |
      echo "alias python3='/usr/bin/python'" >> .bashrc

  - name: Deploying a motd banner
    shell: |
      touch /etc/motd
      echo "{{motd_var}}" >> /etc/motd

  - name: Creating user in both servers
    ansible.builtin.user:
      name: "{{name_var}}"
      state: "{{state_var}}"
      shell: "{{shell_var}}"
      system: "{{system_var}}"
      createhome: "{{createhome_var}}"
      home: "{{home_var}}"
```

**Figure 1.10. The screenshot above shows the contents of config.yml.**

```
[ubuntu]
server1 ansible_connection=ssh ansible_user=piolo javaopenjdk=openjdk-8-jdk
[CentOs]
centos_gui ansible_connection=ssh ansible_user=piolo javaopenjdk=java-1.8.0-openjdk
```

Figure 1.11. The screenshot above shows the contents of the inventory file.

```
piolo@workstation:~/Torrecampo_PrelimExam$ ansible-playbook --ask-become-pass config.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [server1]
ok: [centos_gui]

TASK [Installing python3, pip3 and openjdk] ************************************************************************
ok: [server1]
ok: [centos_gui]

TASK [Setting python3 interpreter as default] **********************************************************************
changed: [server1]
changed: [centos_gui]

TASK [Deploying a motd banner] *************************************************************************************
changed: [server1]
changed: [centos_gui]

TASK [Creating user in both servers] *******************************************************************************
ok: [server1]
ok: [centos_gui]

PLAY RECAP *********************************************************************************************************
centos_gui                 : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server1                    : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

piolo@workstation:~/Torrecampo_PrelimExam$
```

Figure 1.12. The screenshot above shows the result of the ansible playbook.

## VERIFICATION OF TASKS

| Verification of Installed Package | | |
|---|---|---|
| **Package Name** | **Ubuntu Server** | **CentOSServer** |
| **python3** | `piolo@server1:~$ python3 --version`<br>`Python 3.10.4`<br>`piolo@server1:~$` | `[piolo@localhost ~]$ python3 --version`<br>`Python 3.6.8` |
| **python3-pip** | `piolo@server1:~$ python3 -m pip --version`<br>`pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)`<br>`piolo@server1:~$ python3 -m pip --version` | `[piolo@localhost ~]$ python3 -m pip --version`<br>`pip 9.0.3 from /usr/lib/python3.6/site-packages (python 3.6)` |
| **Java OpenJDK** | `piolo@server1:~$ java -version`<br>`openjdk version "1.8.0_342"`<br>`OpenJDK Runtime Environment (build 1.8.0_342-8u342-b07-0ubuntu1~22.04-b07)`<br>`OpenJDK 64-Bit Server VM (build 25.342-b07, mixed mode)`<br>`piolo@server1:~$` | `[piolo@localhost ~]$ java -version`<br>`openjdk version "1.8.0_342"`<br>`OpenJDK Runtime Environment (build 1.8.0_342-b07)`<br>`OpenJDK 64-Bit Server VM (build 25.342-b07, mixed mode)`<br>`[piolo@localhost ~]$` |

Table 1.1. The table above shows the screenshot of package verification on both servers.

| Verifying aliases under .bashrc | |
|---|---|
| **Ubuntu Server** | **CentOS Server** |
| `piolo@server1:~$ cat .bashrc \| grep python3`<br>`alias python3='/usr/bin/python'`<br>`piolo@server1:~$` | `[piolo@localhost ~]$ cat .bashrc \| grep python`<br>`alias python3='/usr/bin/python'`<br>`[piolo@localhost ~]$` |

Table 1.2. The table above shows the screenshot of alias verification on both servers.

| Verifying the MOTD Banner | |
|---|---|
| **Ubuntu Server** | **CentOS Server** |
| piolo@server1:~$ cat /etc/motd<br>Amsible Managed node by Piolo<br>piolo@server1:~$ | [piolo@localhost ~]$ cat /etc/motd<br>Amsible Managed node by Piolo<br>[piolo@localhost ~]$ |

Table 1.3. The table above shows the screenshot of MOTO verification on both servers.

| Verifying Added User | |
|---|---|
| **Ubuntu Server** | **CentOS Server** |
| piolo@server1:~$ cat /etc/passwd \| grep papzi<br>papzi:x:1001:1001::/home/papzi:/bin/bash<br>piolo@server1:~$ | [piolo@localhost ~]$ cat /etc/passwd \| grep papzi<br>papzi:x:1001:1001::/home/papzi:/bin/bash<br>[piolo@localhost ~]$ |

Table 1.4. The table above shows the screenshot of verification of added user on both servers.

5. PUSH and COMMIT your PrelimExam in your GitHub repo

```
piolo@workstation:~/Torrecampo_PrelimExam$ git add *
piolo@workstation:~/Torrecampo_PrelimExam$ git commit -m "Prelim"
[main b6a4602] Prelim
 1 file changed, 5 insertions(+), 2 deletions(-)
piolo@workstation:~/Torrecampo_PrelimExam$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 457 bytes | 457.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:piolotorrecampo/Torrecampo_PrelimExam
   ec5fa77..b6a4602  main -> main
piolo@workstation:~/Torrecampo_PrelimExam$
```

Figure 1.13. Performing the steps of pushing a repository to the Github.

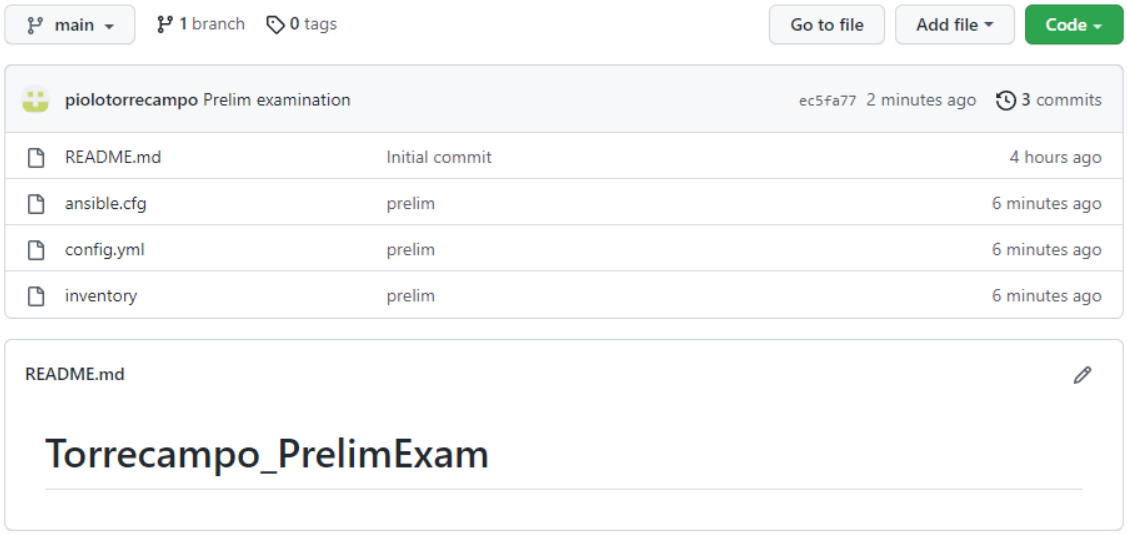


**Figure 1.14. The screenshot above shows the web page of the updated repository.**

6. Your document report should be submitted here.

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

**Github Link:** piolotorrecampo/Torrecampo_PrelimExam (github.com)

**Honor Pledge**

*"I affirm that I will not give or receive unauthorized help on this activity and that all will be my own."*