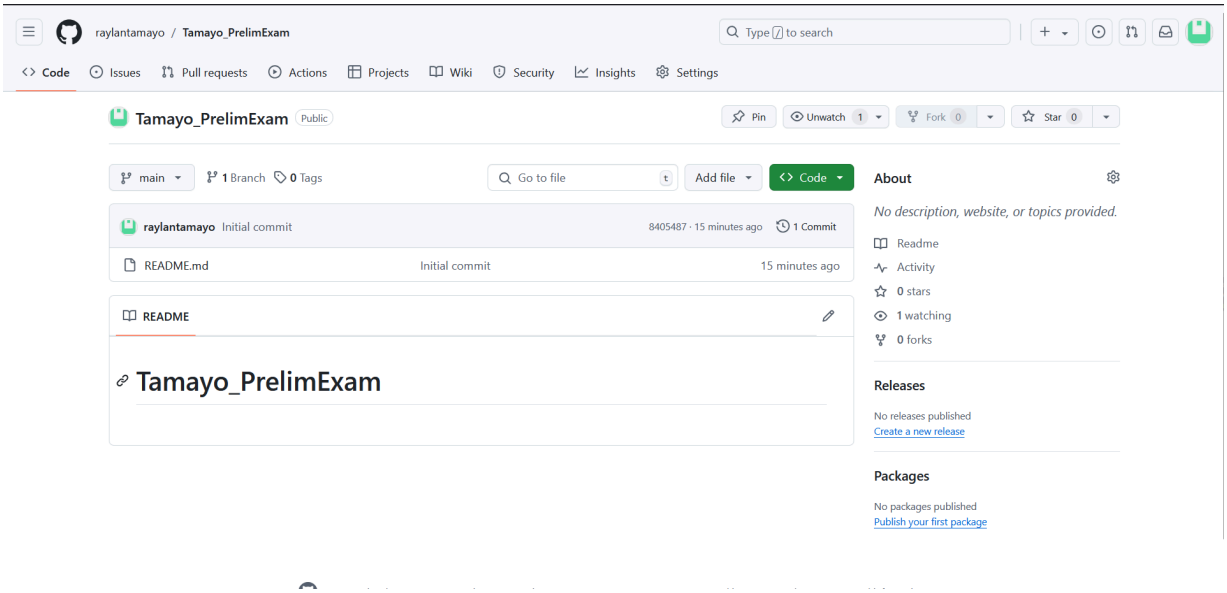
PRELIM EXAM	
Name: Tamayo, Ray Lan A.	**Date:** 09/27/2024
Course/Section: CPE 212-CPE31S21	**Instructor:** Engr. Robin Valenzuela

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



I made a new repository in this part of this activity and in order to clone it in my workstation. I copied the ssh of this repository.

3. Clone your new repository in your CN.

```
tamayo@workstation:~$ git clone git@github.com:raylantamayo/Tamayo_PrelimExam.g
it
Cloning into 'Tamayo_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.
tamayo@workstation:~$
```

In this step, I clone my new repository in my Control Node.

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

```
tamayo@workstation:~/Tamayo_PrelimExam$ touch ansible.cfg inventory
tamayo@workstation:~/Tamayo_PrelimExam$ ls
ansible.cfg  inventory  README.md
tamayo@workstation:~/Tamayo_PrelimExam$
```

tamayo@workstation: ~/Tamayo_PrelimExam

File Edit View Search Terminal Help

GNU nano 2.9.3 inventory

```
192.168.56.128 ansible_python_interpreter=/usr/bin/python3
192.168.56.128 apache_package=apache2 php_package=libapache2

192.168.56.129 ansible_python_interpreter=/usr/bin/python3
192.168.56.129 apache_package=apache2 php_package=libapache2

192.168.56.130 ansible_python_interpreter=/usr/bin/python3
192.168.56.130 apache_package=httpd php_package=php
```

tamayo@workstation: ~/Tamayo_PrelimExam

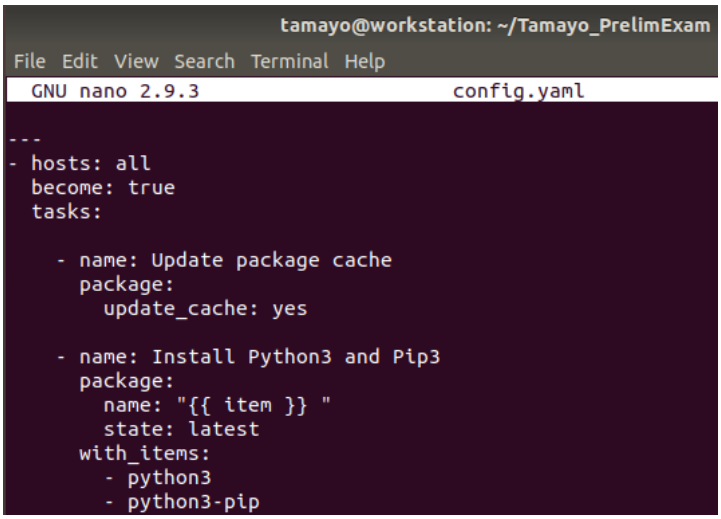
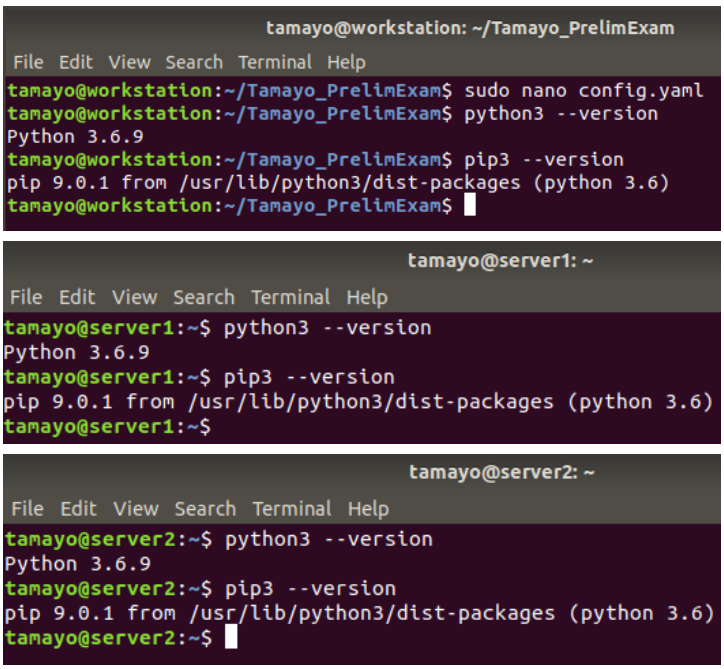
File Edit View Search Terminal Help

GNU nano 2.9.3 ansible.cfg

```
[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings = False

remote_user = tamayo
private_key_file = ~/.ssh/
ask_become_pass = true
```

In this part, I created two files using the touch command. In the inventory, I stored all the IP Addresses of workstation and other servers. While in the ansible, I stored the script from the previous HOA so that it wont be time consuming.

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

INPUT	
OUTPUT	

In this step, we install the latest python3 and pip by using a playbook which is config.yaml

- use pip3 as default pip

INPUT	`- name: Set pip3 as the default python for workstation`
	`  command: "ln -sf /usr/bin/pip3 /usr/bin/pip"`
	`  when: ansible_distribution == "workstation"`
	
	`- name: Set pip3 as the default python for server1`
	`  command: "ln -sf /usr/bin/pip3 /usr/bin/pip"`
	`  when: ansible_distribution == "server1"`
	
	`- name: Set pip3 as the default python for server2`
	`  command: "ln -sf /usr/bin/pip3 /usr/bin/pip"`
	`  when: ansible_distribution == "server2"`

OUTPUT	`tamayo@workstation:~/Tamayo_PrelimExam$ which pip3`
	`/usr/bin/pip3`
	`tamayo@workstation:~/Tamayo_PrelimExam$`
	
	`tamayo@server1:~$ which pip3`
	`/usr/bin/pip3`
	`tamayo@server1:~$`
	
	`tamayo@server2:~$ which pip3`
	`/usr/bin/pip3`
	`tamayo@server2:~$`

In this step, we use pip3 as default pip by using the command above.

- use python3 as default pip

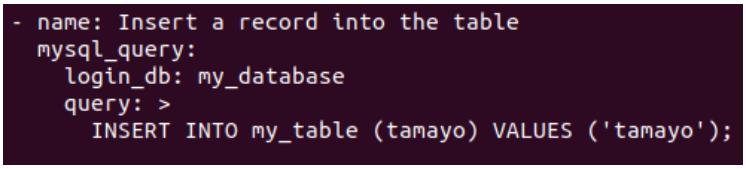
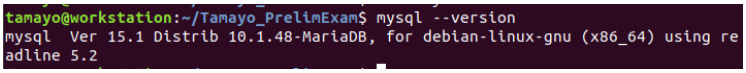
INPUT	`- name: Set python3 as the default python for workstation`
	`  command: "ln -sf /usr/bin/python3 /usr/bin/python"`
	`  when: ansible_distribution == "workstation"`
	
	`- name: Set python3 as the default python for server1`
	`  command: "ln -sf /usr/bin/python3 /usr/bin/python"`
	`  when: ansible_distribution == "server1"`
	
	`- name: Set python3 as the default python for server2`
	`  command: "ln -sf /usr/bin/python3 /usr/bin/python"`
	`  when: ansible_distribution == "server2"`

OUTPUT	`tamayo@workstation:~/Tamayo_PrelimExam$ which python3`
	`/usr/bin/python3`
	`tamayo@workstation:~/Tamayo_PrelimExam$`
	
	`tamayo@server1:~$ which python3`
	`/usr/bin/python3`
	`tamayo@server1:~$`
	
	`tamayo@server2:~$ which python3`
	`/usr/bin/python3`
	`tamayo@server2:~$`

In this step, we use python3 as default pip as command shown above.

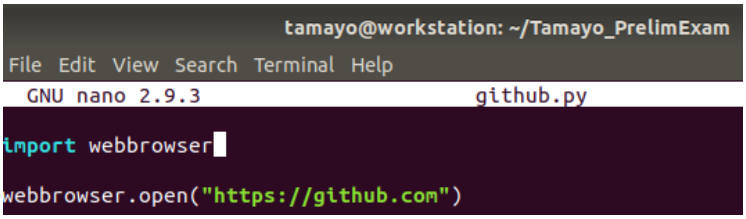
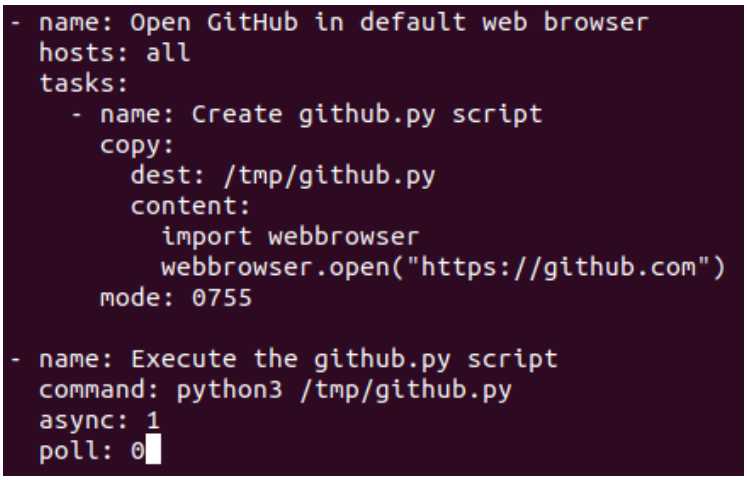
- Install MariaDB as well as starting the server, create a database and a table using mariaDB and input one record into a table USING ANSIBLE ONLY

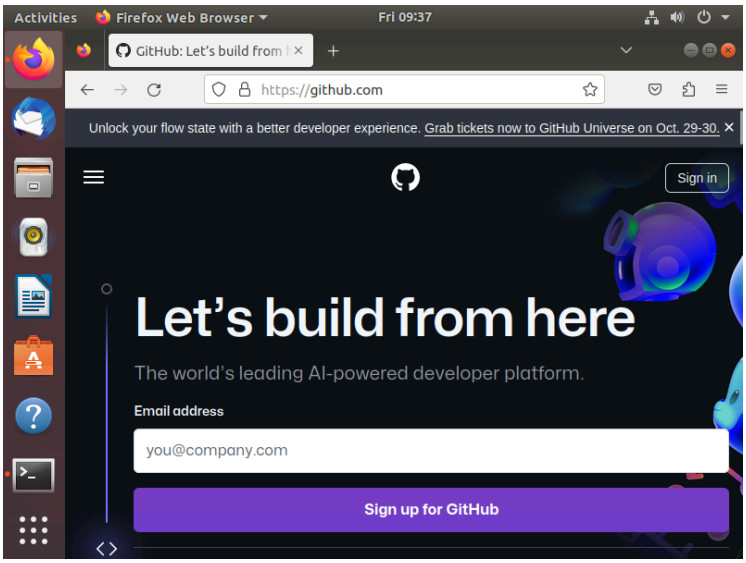
INPUT	```
	- name: Setup MariaDB
	  hosts: mariadb_servers
	  become: yes
	  tasks:
	
	- name: Install MariaDB
	  apt:
	  name: mariadb-server
	  state: present
	  when: ansible_os_family == "Debian"
	
	- name: Start MariaDB service
	  service:
	  name: mariadb
	  state: started
	  enabled: yes
	``
	``
	- name: Create database
	  mysql_db:
	  name: my_database
	  state: present
	
	- name: Create table
	  mysql_db:
	  name: my_database
	  state: present
	  collation: utf8_general_ci
	  encoding: utf8
	``
	``
	- name: Create a table in the database
	  mysql_query:
	  login_db: my_database
	  query: >
	  CREATE TABLE IF NOT EXISTS my_table (
	  id INT AUTO_INCREMENT PRIMARY KEY,
	  name VARCHAR(255) NOT NULL
	  );
	``

	
OUTPUT	

In this step, we install MariaDB as well as starting the server, as shown on the table.

- Create a Python Script named "github.py" that automatically opens github on Firefox (or any default web browser) USING ANSIBLE.

INPUT	 

OUTPUT	

In this task, we created a python script that automatically opens a github on firefox.

- 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)"

INPUT	

```
- name: Set up MOTD
  hosts: all
  become: yes
  vars_files:

- config.yaml
  tasks:

- name: Create MOTD file
  copy:
    dest: /etc/motd
    content: "{{ motd_message | default('Ansible Managed node by ' + tamay$
```

In this task, we created an Motd that contains the following text.

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

INPUT	

```
- name: Create a new user
  hosts: all
  become: yes
  vars_files:

- config.yaml
  tasks:

- name: Create a user
  user:
    name: "{{ tamayo }}"
    password: "{{ raylansysad2 | password_hash('sysad2') }}"
    state: present
```

In the final task, we created a script that creates a user with a variable defined in config.yaml

5. PUSH and COMMIT your PrelimExam in your GitHub repo

```
tamayo@workstation:~/Tamayo_PrelimExam$ git config --global user.email "raylant
amayo@gmail.com"
tamayo@workstation:~/Tamayo_PrelimExam$ git config --global user.name "tamayo"
tamayo@workstation:~/Tamayo_PrelimExam$ git commit -m "Tamayo Prelim Exam"
[main 999f2bb] Tamayo Prelim Exam
 5 files changed, 153 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 config.retry
 create mode 100644 config.yaml
 create mode 100644 github.py
 create mode 100644 inventory
tamayo@workstation:~/Tamayo_PrelimExam$ git push origin
Counting objects: 7, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.64 KiB | 1.64 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To github.com:raylantamayo/Tamayo_PrelimExam.git
   8405487..999f2bb  main -> main
tamayo@workstation:~/Tamayo_PrelimExam$
```

In this task, I command the git config to access my github and commit it into my repository, also pushing it through using git push origin.

6. Screenshot / document your work. Add PROOF that all your CODES / SCRIPTS WORK.

7. Your document report should be submitted here. Your document SHOULD BE explained neatly and comprehensively.

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

https://github.com/raylantamayo/Tamayo_PrelimExam.git