Name: Renier L. Lope	**Date Performed: 12/04/2024**
Course/Section: CPE31S2	**Date Submitted: 12/04/2024**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st Sem (2024-2025)**

Final Skills Exam: Hands-on Final Exam

1. Tools Needed

1. VM with Ubuntu, CentOS and Ansible installed
2. Web browser

2. Instructions

1. Create a repository and label it as "Final_Exam_Surname"

2. Clone your new repository in your VM

3. Create an Ansible playbook that does the following with an input of a config.yaml file and structure inventory file.

3.1 Install and configure one enterprise service that can be installed in Debian and Centos servers

3.2 Install and configure one monitoring tool that can be installed in Debian and Centos servers (if it is a stack there should be option of different host)

4.4 Change Motd as "Ansible Managed by <username>"

4. Push and commit your files in GitHub

5. Make sure to show evidence of input (codes) process (codes successfully running) and output (evidence of installation)

5. For your final exam to be counted, please paste your repository link as an answer in this exam.

Note: Extra points if you will implement the said services via containerization.

Output:

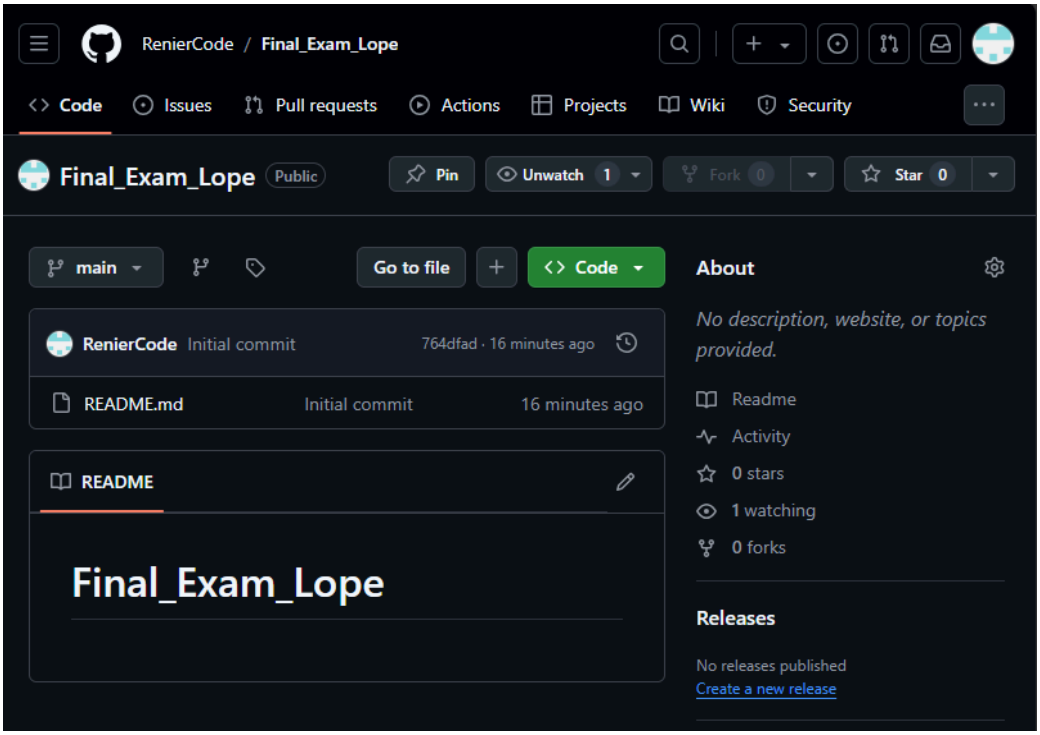

Figure 1: Create a repository and label it as "Final_Exam_Lope".

```
rnrlope@workstation:~$ git clone git@github.com:RenierCode/Final_Exam_Lope
Cloning into 'Final_Exam_Lope'...
Warning: Permanently added the ECDSA host key for IP address '4.237.22.38' to th
e list of known hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
Receiving objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
rnrlope@workstation:~$ cd Final_Exam_Lope
rnrlope@workstation:~/Final_Exam_Lope$
```

Figure 2: Clone your new repository in your VM.

```
rnrlope@workstation:~/Final_Exam_Lope$ nano ansible.cfg
rnrlope@workstation:~/Final_Exam_Lope$ cat ansible.cfg
[defaults]
inventory = inventory
remote_user = rnrlope
host_key_checking = True
private_key_file = ~/.ssh/ansible
deprecation_warnings = False
rnrlope@workstation:~/Final_Exam_Lope$ nano inventory
rnrlope@workstation:~/Final_Exam_Lope$ cat inventory
[debian_server]
server1

[centos_server]
centOS
rnrlope@workstation:~/Final_Exam_Lope$
```

Figure 3: Create the ansible.cfg and inventory file.

```
rnrlope@workstation:~/Final_Exam_Lope$ nano config.yml
rnrlope@workstation:~/Final_Exam_Lope$ cat config.yml
---
- hosts: all
  become: true
  pre_tasks:

  - name: update repository index (CentOS)
    tags: always
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"

  - name: install updates (Ubuntu)
    tags: always
    apt:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

- hosts: all
  become: true
  roles:
     - base

- hosts: debian_server:centos_server
  become: true
  roles:
     - docker
     - nagios
     - motd
rnrlope@workstation:~/Final_Exam_Lope$
```

Figure 3: Create a playbooks file named “config.yml”. Inside create a task to automate the update of the repository index and also create the prompt to use roles.

```
rnrlope@workstation:~/Final_Exam_Lope$ mkdir roles
rnrlope@workstation:~/Final_Exam_Lope$ cd roles
rnrlope@workstation:~/Final_Exam_Lope/roles$ mkdir base docker nagios
rnrlope@workstation:~/Final_Exam_Lope/roles$ cd base
rnrlope@workstation:~/Final_Exam_Lope/roles/base$ mkdir tasks
rnrlope@workstation:~/Final_Exam_Lope/roles/base$ cd ..
rnrlope@workstation:~/Final_Exam_Lope/roles$ cd docker
rnrlope@workstation:~/Final_Exam_Lope/roles/docker$ mkdir tasks
rnrlope@workstation:~/Final_Exam_Lope/roles/docker$ cd ..
rnrlope@workstation:~/Final_Exam_Lope/roles$ cd nagios
rnrlope@workstation:~/Final_Exam_Lope/roles/nagios$ mkdir tasks
rnrlope@workstation:~/Final_Exam_Lope/roles/nagios$ cd ..
rnrlope@workstation:~/Final_Exam_Lope/roles$
```

Figure 4: Create a new directory named “roles” to store the roles and inside, create directories for the specified roles such as “base”, “docker”, and “nagios” to separate the tasks based on the groups in inventory.

```
rnrlope@workstation:~/Final_Exam_Lope$ cd roles/base/tasks
rnrlope@workstation:~/Final_Exam_Lope/roles/base/tasks$ nano main.yml
rnrlope@workstation:~/Final_Exam_Lope/roles/base/tasks$ cat main.yml
---
- name: install updates (CentOS)
  tags: always
  dnf:
    update_only: yes
    update_cache: yes
  when: ansible_distribution == "CentOS"

- name: install updates (Ubuntu)
  tags: always
  apt:
    upgrade: dist
    update_cache: yes
  when: ansible_distribution == "Ubuntu"
rnrlope@workstation:~/Final_Exam_Lope/roles/base/tasks$
```

Figure 5: Create a playbook file inside ‘roles/base/tasks’. This will install updates for both the Ubuntu and CentOS servers.

```
rnrlope@workstation:~/Final_Exam_Lope$ cd roles/docker/tasks
rnrlope@workstation:~/Final_Exam_Lope/roles/docker/tasks$ nano main.yml
rnrlope@workstation:~/Final_Exam_Lope/roles/docker/tasks$ cat main.yml
---
- name: Install Docker (CentOS)
  dnf:
    name: docker
    state: latest
  when: ansible_distribution == "CentOS"

- name: Install Docker (Ubuntu)
  apt:
    name: docker.io
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Start Docker Service
  service:
    name: docker
    state: restarted
    enabled: yes

- name: Add docker group to current user
  group:
    name: docker
    state: present

- name: Add user to docker group
  user:
    name: rnrlope
    groups: docker
    append: yes
```

```
- name: change permission of docker.sock
  file:
    path: /var/run/docker.sock
    state: file
    owner: root
    group: docker
    mode: "666"

rnrlope@workstation:~/Final_Exam_Lope/roles/docker/tasks$
```

Figure 6 - 7: Create a playbook file inside ‘roles/base/tasks’. This will install and start Docker Service for both the Ubuntu and CentOS servers.

```
rnrlope@workstation:~/Final_Exam_Lope$ cd roles/nagios/tasks
rnrlope@workstation:~/Final_Exam_Lope/roles/nagios/tasks$ nano main.yml
rnrlope@workstation:~/Final_Exam_Lope/roles/nagios/tasks$ cat main.yml
---
- name: Install nagios requirements (Ubuntu)
  apt:
    name:
      - autoconf
      - gcc
      - libc6
      - make
      - wget
      - unzip
      - libssl-dev
      - apache2
      - php
      - libapache2-mod-php7.2
      - libgd-dev
      - build-essential
    state: latest
    force: true
    update_cache: true
  become: true
  when: ansible_distribution == "Ubuntu"
```

```
- name: Install nagios requirements (CentOS)
  dnf:
    name:
      - gcc
      - glibc
      - glibc-common
      - wget
      - unzip
      - httpd
      - php
      - gd
      - gd-devel
      - perl
      - postfix
      - openssl-devel
    state: latest
  become: true
  when: ansible_distribution == "CentOS"

- name: Install EPEL repo (CentOS)
  dnf:
    name: epel-release
    state: latest
  when: ansible_distribution == "CentOS"

- name: Install nagios (CentOS)
  dnf:
    name: nagios
    state: latest
  when: ansible_distribution == "CentOS"
```

```
- name: Install nagios3 (Ubuntu)
  apt:
    name: nagios3-core
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: Start and Enable Nagios (CentOS)
  service:
    name: nagios
    state: restarted
    enabled: true
  when: ansible_distribution == "CentOS"

- name: Start and Enable Nagios (Ubuntu)
  service:
    name: nagios3
    state: restarted
    enabled: true
  when: ansible_distribution == "Ubuntu"
rnrlope@workstation:~/Final_Exam_Lope/roles/nagios/tasks$
```

Figure 8 - 10: Create a playbook file inside 'roles/base/tasks'. This will install and start Nagios an Event Monitoring tool for both the Ubuntu and CentOS servers.

```
rnrlope@workstation:~/Final_Exam_Lope$ nano inventory
rnrlope@workstation:~/Final_Exam_Lope$ cat inventory
[debian_server]
server1  ansible_user=rnrlope

[centos_server]
centOS   ansible_user=rnrlope
rnrlope@workstation:~/Final_Exam_Lope$
```

Figure 11: Edit the inventory file to include the variable ansible_user.

```
rnrlope@workstation:~/Final_Exam_Lope$ cd roles
rnrlope@workstation:~/Final_Exam_Lope/roles$ mkdir motd
rnrlope@workstation:~/Final_Exam_Lope/roles$ cd motd
rnrlope@workstation:~/Final_Exam_Lope/roles/motd$ mkdir tasks
rnrlope@workstation:~/Final_Exam_Lope/roles/motd$ cd tasks
rnrlope@workstation:~/Final_Exam_Lope/roles/motd/tasks$ nano main.yml
rnrlope@workstation:~/Final_Exam_Lope/roles/motd/tasks$ cat main.yml
---
- name: Ensure sshd is enabled
  service:
    name: sshd
    state: restarted
    enabled: yes

- name: Set MOTD
  lineinfile:
    path: /etc/motd
    line: "Ansible Managed by {{ansible_user}}"
rnrlope@workstation:~/Final_Exam_Lope/roles/motd/tasks$
```

Figure 12: Create another role for Setting MOTD.

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

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

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

TASK [update repository index (CentOS)] ****************************************
skipping: [server1]
ok: [centOS]

TASK [install updates (Ubuntu)] ************************************************
skipping: [centOS]
ok: [server1]

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

Figure 13: Execute the playbook 'config.yml'.

```
TASK [Install nagios (CentOS)] *************************************************
skipping: [server1]
ok: [centOS]

TASK [Install nagios3 (Ubuntu)] ************************************************
skipping: [centOS]
ok: [server1]

TASK [nagios : Start and Enable Nagios (CentOS)] *******************************
skipping: [server1]
changed: [centOS]

TASK [nagios : Start and Enable Nagios (Ubuntu)] *******************************
skipping: [centOS]
changed: [server1]

TASK [motd : Ensure sshd is enabled] *******************************************
changed: [server1]
changed: [centOS]

TASK [motd : Set MOTD] *********************************************************
changed: [server1]
ok: [centOS]

PLAY RECAP *********************************************************************
centOS                     : ok=16   changed=4    unreachable=0    failed=0    skipp
ed=6    rescued=0    ignored=0
server1                    : ok=15   changed=4    unreachable=0    failed=0    skipp
ed=7    rescued=0    ignored=0

rnrlope@workstation:~/Final_Exam_Lope$
```

Figure 14: Play Recap of the playbook 'config.yml'. As we can see here, all of my plays are successful.

Verify Status:

```
rnrlope@server1:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Wed 2024-12-04 09:46:12 +08; 2min 20s ago
     Docs: https://docs.docker.com
 Main PID: 15571 (dockerd)
    Tasks: 7
   CGroup: /system.slice/docker.service
           └─15571 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/conta
```

```
[rnrlope@localhost ~]$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor prese
t: disabled)
   Active: active (running) since Tue 2024-12-03 20:46:14 EST; 8min ago
     Docs: http://docs.docker.com
 Main PID: 11986 (dockerd-current)
    Tasks: 22
   CGroup: /system.slice/docker.service
           ├─11986 /usr/bin/dockerd-current --add-runtime docker-runc=/usr/li...
           └─11992 /usr/bin/docker-containerd-current -l unix:///var/run/dock...

Dec 03 20:46:12 localhost.localdomain dockerd-current[11986]: time="2024-12-0...
Dec 03 20:46:13 localhost.localdomain dockerd-current[11986]: time="2024-12-0...
Dec 03 20:46:13 localhost.localdomain dockerd-current[11986]: time="2024-12-0...
Dec 03 20:46:13 localhost.localdomain dockerd-current[11986]: time="2024-12-0...
Dec 03 20:46:14 localhost.localdomain dockerd-current[11986]: time="2024-12-0...
Dec 03 20:46:14 localhost.localdomain dockerd-current[11986]: time="2024-12-0...
Dec 03 20:46:14 localhost.localdomain dockerd-current[11986]: time="2024-12-0...
Dec 03 20:46:14 localhost.localdomain dockerd-current[11986]: time="2024-12-0...
Dec 03 20:46:14 localhost.localdomain dockerd-current[11986]: time="2024-12-0...
```

Figure 15: Verify if Docker Service is successfully installed on both server1 and centOS.

```
rnrlope@server1:~$ systemctl status nagios3
● nagios3.service - LSB: nagios host/service/network monitoring and management
   Loaded: loaded (/etc/init.d/nagios3; generated)
   Active: active (running) since Wed 2024-12-04 09:46:39 +08; 1min 55s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 16303 ExecStop=/etc/init.d/nagios3 stop (code=exited, status=0/SUCCE
  Process: 16325 ExecStart=/etc/init.d/nagios3 start (code=exited, status=0/SUC
    Tasks: 1 (limit: 4541)
   CGroup: /system.slice/nagios3.service
           └─16383 /usr/sbin/nagios3 -d /etc/nagios3/nagios.cfg
```

```
[rnrlope@localhost ~]$ systemctl status nagios
● nagios.service - Nagios Core 4.4.14
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; vendor prese
t: disabled)
   Active: active (running) since Tue 2024-12-03 20:46:38 EST; 7min ago
     Docs: https://www.nagios.org/documentation
  Process: 12778 ExecStopPost=/usr/bin/rm -f /var/spool/nagios/cmd/nagios.cmd (c
ode=exited, status=0/SUCCESS)
  Process: 12775 ExecStop=/usr/bin/kill -s TERM ${MAINPID} (code=exited, status=
0/SUCCESS)
  Process: 12788 ExecStart=/usr/sbin/nagios -d /etc/nagios/nagios.cfg (code=exit
ed, status=0/SUCCESS)
  Process: 12783 ExecStartPre=/usr/sbin/nagios -v /etc/nagios/nagios.cfg (code=e
xited, status=0/SUCCESS)
 Main PID: 12790 (nagios)
    Tasks: 6
   CGroup: /system.slice/nagios.service
           ├─12790 /usr/sbin/nagios -d /etc/nagios/nagios.cfg
           ├─12791 /usr/sbin/nagios --worker /var/spool/nagios/cmd/nagios.qh
           ├─12792 /usr/sbin/nagios --worker /var/spool/nagios/cmd/nagios.qh
           ├─12795 /usr/sbin/nagios --worker /var/spool/nagios/cmd/nagios.qh
           ├─12796 /usr/sbin/nagios --worker /var/spool/nagios/cmd/nagios.qh
           └─12813 /usr/sbin/nagios -d /etc/nagios/nagios.cfg
```

Figure 16: Verify if Nagios Event Monitoring Tool is successfully installed on both server1 and centOS.

```
rnrlope@server1:~$ cat /etc/motd
Ansible Managed by rnrlope
rnrlope@server1:~$
```

```
[rnrlope@localhost ~]$ cat /etc/motd
Ansible Managed by rnrlope
[rnrlope@localhost ~]$
```

Figure 17: Run 'cat /etc/motd' to display if the MOTD is successfully set on both server1 and centOS.

GIT PUSH:

```
rnrlope@workstation:~/Final_Exam_Lope$ git add --all
rnrlope@workstation:~/Final_Exam_Lope$ git commit -m "FINAL EXAM"
[main 52128d2] FINAL EXAM
 7 files changed, 176 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 config.yml
 create mode 100644 inventory
 create mode 100644 roles/base/tasks/main.yml
 create mode 100644 roles/docker/tasks/main.yml
 create mode 100644 roles/motd/tasks/main.yml
 create mode 100644 roles/nagios/tasks/main.yml
rnrlope@workstation:~/Final_Exam_Lope$ git push origin main
Counting objects: 18, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (18/18), 2.18 KiB | 2.18 MiB/s, done.
Total 18 (delta 0), reused 0 (delta 0)
To github.com:RenierCode/Final_Exam_Lope
   764dfad..52128d2  main -> main
rnrlope@workstation:~/Final_Exam_Lope$
```

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